uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

u Ottawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Gustavo Frederico

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.C.S. (Master of Computer Science)

GRADE / DEGREE

The School of Information Technology and Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Feature Selection and Evaluation for Genre Classification of Symbolically Encoded Classical Music with the Aid of Machine Learning

TITRE DE LA THÈSE / TITLE OF THESIS

Won Sook Lee

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

Nathalie Japkowicz

John Oommen

Gary W. Slater

Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

FEATURE SELECTION AND EVALUATION FOR GENRE CLASSIFICATION OF SYMBOLICALLY ENCODED CLASSICAL MUSIC WITH THE AID OF MACHINE LEARNING

Gustavo Cesar de Souza Frederico

Thesis

submitted to the Faculty of Graduate and Postdoctoral Studies

in partial fulfillment of the requirements for the degree of Master of Computer Science

June 1st, 2006

Ottawa-Carleton Institute for Computer Science

School of Information Technology and Engineering

University of Ottawa

Your file Votre référence
ISBN: 978-0-494-18417-2
Our file Notre référence
ISBN: 978-0-494-18417-2

Abstract

This work defines useful features for the classification of symbolically encoded music into 14 classical genres namely chorale, symphony, étude, fugue, prelude, contrafactum, sonata, mazurka, motet, sonatina, waltze, concerto, Gregorian chant and scherzo. Features are based on Music Theory and grouped into seven categories: distances in the harmonic möbius strip, distances on the line of fifths, scale, rhythmic syncopation and meter, polyphony measurements, duration and instrumentation. Features are extracted and ranked combining 5 filter-based methods. Six Machine Learning algorithms are defined for classification: three Support Vector Machines, one Bayesian network, the C4.5 and random forests. Using nested cross-validation for training and testing and considering all the features, the Bayesian network classifier yields 84.10 % empirical accuracy. The FEATUROMETRE process measures the usefulness of the feature subsets in an approach similar to wrapper methods, conveying relevant information to domain experts. Another experiment measures the usefulness and accuracy of features individually and by category using FEATUROMETRE. Grouping the music pieces by their period, the measured accuracy with the random forest classifier in the second experiment reaches 89.81 %.

Acknowledgements

I would like to thank my wife Louise for her support and understanding throughout my Master's program. *Muito obrigado*. I would like to thank my good friend Sidney Givigi Junior for his academic guidance. *Muito obrigado*. I would like to thank my parents Monfardini Frederico and professor Denise Frederico for teaching me music. Thanks also to professor Denise Frederico for her comments on the musical taxonomy used in this work. *Muito obrigado*. I would like to thank Professor WonSook Lee for her supervision and encouragement throughout this research. I would like to thank Professor Mario Marchand for his supervision in the beginning of this research. My gratitude also goes to MTS Allstream for the logistical support of this research. I would like to thank my friend Mohak Shah for his help at various occasions. I am also thankful to my friend Steven Bergman for reviewing this text. Thanks to my colleague Pengcheng Xi for the Weka overview.

Table of Contents

Table of Figures

List of Tables

Chapter 1 Introduction

1.1 Motivation

Genre is one classification system that allows us to organize and refer to groups of musical works. It is relatively easy for the musically untrained person to distinguish between broad musical genres such as jazz, rock and classical. This ability is one of many complex tasks that human beings do well. We can utilize our innate hearing ability to notice fine details in sound, while associating past experiences with new ones, based on complex perceptions of the environment and context. The notion of genre depends on the associations of musical works among themselves according to certain similarities.

The branches of knowledge that study music have existed for centuries. Throughout history, musicology, music theory, composition and music analysis evolved in the understanding of musical works. These branches of knowledge produced rich and plentiful notions and definitions. The notion of genre, even if not formally defined, provided enough clarity for the understanding and organization of the musical works. For instance, musical analysis tries to describe certain characteristics of segments of a piece, whole pieces or a collection of pieces of music. It utilizes the definitions of music theory, sometimes with formal terms, and other times with subjective ones. An example of a formal term would include chord labels in a Bach chorale. This and other examples describe analytical annotations that can be automatically derived from the contents of the piece. At times, however, musical analysts work with subjective concepts. In comparing criticism with musical analysis, Bent comments on the fact that analysis often uses subjective concepts:

> "In general, [musical] analysis is more concerned with describing than with judging. In this sense, analysis goes less far than criticism, and it does so essentially because it aspires to objectivity and considers judgment to be subjective. [...] the analyst's definable elements (a phrase, a motif etc.) are often defined by subjective conditions. Where subjectivities are acknowledged to be inevitable, the analytical

> mind will tend not to work with them directly, but to investigate their nature in relation to definable musical phenonema, thus drawing closer to aesthetics in general and to semiology in particular." (Bent & Pople, 2006)

In the present work, we intend to automatically learn and classify classical music into genres with the aid of machine learning. The genres of the pieces in our dataset have the historic labels commonly known in the branches of music. Therefore, we inherit the underlying taxonomy and along with it its subjective semantics. When we classify one piece of music into one genre or another we are in fact also putting the traditional taxonomy and semantics to the test. If on the one hand ambiguity may lower optimal classification accuracy, on the other we are making an experiment that can augment the understanding of the problem domain with proper analysis. We have, thus, the opportunity to create new tools for musicologists to characterize genres. If in the computer system's model we abstract the musical objects and share the vocabulary with the traditional music sciences, we will further enable a joint research of genre characteristics.

It is necessary to make the distinction between the classification into musical genres and another problem found in the field of Music Information Retrieval, namely that of matching one music instance against a musical database. In the later, one music instance is often given by a user and is potentially inaccurate in its execution. One example of such problem is query by humming. The goals in query by humming are the accuracy of the matches as well as the performance of the algorithms. Both problems depend on the comparison of musical pieces in general. Classification into musical genres, however, does not match one music instance against a database for identification, but rather assigns the label of a genre to the music instance. The instance may have an entirely different authorship and melody from all other instances used during the learning process and may still be correctly classified.

Genre can be identified by humans not only through access to historical information such as composer and epoch, but also through some complex perceptual processes that recognize characteristics and structure inherent to the piece, including rhythm, scale, melody, harmony and timbre. These characteristics are then compared against other

works that the listener has already heard. However, how can computers determine the genre of a musical piece?

Machine Learning is the branch of Artificial Intelligence that develops techniques to allow computer systems to learn from an underlying problem domain. Machine Learning algorithms have been successfully used in many pattern recognition domains. A few examples include medical diagnosis, handwriting recognition, face recognition, time series prediction, text classification, identification of performers by their playing style and natural language processing. Only relatively recently has work in the field of Information Retrieval specialized in Music Information Retrieval and genre classification with the use of Machine Learning. Some factors that challenge research of this problem include the difficulty of access to large, representative datasets, inherent subjectivity of the traditional genre taxonomy, relatively limited multidisciplinary research involving Music and Computer Science and the lack of standardized taxonomies for benchmarks. The last two factors make the fair comparison of the existing works difficult. Also, because music pieces are complex objects and Music Information Retrieval addresses different problems, systems adopt different representations of the data. Specifically, genre classification systems have used different feature sets for data representation. Previous works, however, have not documented the relationships between the data representation and system performance. The current work tries to address the problem proposing a novel procedure.

Despite the complexity inherent of the musical objects and different approaches to model them, computing in music has been successfully performed to a certain extent. We now observe more specifically other works of music classification.

1.2 State of the Art of Music Classification

Machine Learning has been used in genre classification at experimentations in various settings. Previous works on the problem have utilized two main approaches for data representation: sound recordings and symbolic data. Works with sound recordings take advantage of the ease of accessibility to very large datasets, not requiring profound domain knowledge in order to achieve satisfactory classification accuracy. Symbolic

musical representation arguably provides a semantically richer set of features that allow the high-level manipulation of the musical objects. The algorithms employed so far in the task varied. Examples include Support Vector Machines (SVMs), custom algorithms, k-Nearest Neighbour, and Bayes classifiers.

In this section we review previous works in the area. The results from previous works are encouraging, even though most of works so far have utilized either broad genres such as classical, rock, pop and jazz or use small subsets such as Folkloric European genres. No work so far has studied the classification of symbolically encoded music into classical genres with the aid of Machine Learning algorithms. Experiments of genre recognition with humans have been carried out. In one experiment, people are able to distinguish between ten broad genres with an accuracy of 71.7 % (Perrot & Gjerdigen, 1999) after hearing 300 milliseconds of audio. The ten genres were country, jazz, pop, R&B, rock, rap, blues, classical, dance and latin. This accuracy is below what has been achieved in automated genre classification in broad taxonomies, even though taxonomies in the experiments with humans and machines do not exactly coincide.

Previous work of music genre classification can be categorized into sound-based and symbolic classification research.

1.2.1 Sound-based Research

One referential sound-based work is that of Tzanetakis and Cook (Tzanetakis & Cook, 2002). They define three feature sets for genre categorization: timbral texture, rhythmic content and pitch content[1]. In total 30 features are defined. In an experiment three classifiers are compared: the simple Gaussian, the Gaussian mixture model and the K-nearest neighbour. Multiple datasets are tested individually. The musical dataset contains

---

[1] The expressions "instrumentation" and "timbral texture" are utilized as synonyms to designate the first feature set in the article. Theoretically, however, "instrumentation" refers to the choice of instruments for a certain piece and is at times used as synonym of "orchestration". We would argue that the term "timbral texture" is more appropriate in this case, considering the kinds of processing applied to the sounds.

10 genres: classical, country, disco, hiphop, jazz, rock, blues, reggae, pop and metal. The classical dataset had four subgenres: choir, orchestra, piano and string quartet. A discussion regarding the choice of the particular classical subgenres is at the conclusion of this text. An accuracy of 61% in an experiment is reported. Sound-based classification by Pye achieved 92% accuracy in a collection of 6 genres: blues, easy listening, classical, opera, trance (techno) and indie rock (Pye, 2000). The exact size of the training dataset is not reported, but the testing set separate from the training set is reported to contain 175 songs. The article also performs a query-by-similarity test. Xu et al. employ Support Vector Machines in a collection of 100 songs that belong to four different genres, reporting an accuracy of 93% (Xu, Maddage, Shao, Cao, & Tian, 2003). Three SVMs are used in hierarchy for training and testing. In the second layer of the hierarchy songs belong to two pairs of clustered genres: pop/classic or rock/jazz. In the third layer of the hierarchy two SVMs classify the songs into one of the four genres. The two reports do not present any comparative analysis of genre characteristics.

1.2.2 Symbolically Encoded Music Research

Using symbolic encoding Chai and Vercoe report 77% accuracy for classification between two genres of folk music (Chai & Vercoe, 2001). They reported an accuracy of 63% using three folkloric genres and a dataset of 491 monophonic pieces. Pitch and duration features were programmatically extracted from files encoded in kern and EsAC (Essen Associative Code) (Schaffrath, 1997). Chai and Vercoe also briefly discuss the influence of melodic, intervallic and rhythmic features in classification accuracy for the specific genres. For instance, the experiments showed that intervallic information performed better than absolute melody contour. Intuitively, that is consistent with the idea that perception is invariant to key modulation.

Shan and Kuo report 64% to 84.2 % accuracy classifying music into four possible styles, each pair at a time (Shan, Kuo, & Chen, 2003). Music pieces belong to four arbitrary styles: Enya, Beatles, Chinese folk songs and Japanese folk songs. In the experiment, the expression "style" is used as a description of a human feeling and the choice of style categories for the experiment is not described in further detail. The exact size of the

dataset is not mentioned, but each genre is said to have between 39 to 55 music pieces. Polyphonic MIDI files are used in the experiment. The system performs melody extraction and a harmonic analysis of the music, generating chord sequences. Two mining algorithms are then used to take the chord sequences as input: frequent itemset and frequent substring. Both methods take into account the frequency of occurrence of chords in the training set. In the frequent substring method, chords are equivalent to strings. Tests are also carried with choruses, which are main portions of melodic lines.

Pérez-Sancho et al. use a text categorization approach to classify music (Pérez-Sancho et al., 2005). They use two datasets for training and testing. The first dataset has 110 non-quantized MIDI files divided between classical and jazz genres. The second dataset has 300 music samples divided into three genres: Gregorian chants, baroque pieces by J.S. Bach and Ragtimes. The dataset included monophonic sequences only. Pitch and duration are translated into strings achieving an accuracy rate of 94.3%. The algorithm is a naïve Bayes classifier.

McKay (C. McKay, 2004)(C. McKay & Fujinaga, 2004) reports an accuracy of 86% in a flat taxonomy with 9 leaf genres. In a taxonomy with 38 leaf genres, the accuracy of 57% is reported. The larger taxonomy has nine root genres namely: country, jazz, modern pop, rap, rhythm and blues, rock, western classical, western folk and worldbeat. Feature selection is performed during training using genetic algorithms to reduce the number of features used during testing. The actual number of selected features at runtime is not reported. McKay's Thesis also mentions in its section of future research that an analysis of the feature's influence in genre classification has musicological merit (C. McKay, 2004). As we shall see, our research takes a similar approach to McKay's, such as in the use of high-level features and the large size of the dataset. Differences, however, include the explicit quantitative measurement of the usefulness of features, learning algorithms employed, feature selection process, classification accuracy result, the hierarchization of the genre taxonomy, and the particular genres used.

Most of the previous works concentrate on broad genre categorization such as jazz, rock, pop and classical. Intuitively, it is more difficult for a lay person to distinguish subgenres than to distinguish between broad genres. That is because an accurate discernment of

more specialized genres requires finer perception of more subtle details. Thus, the genre taxonomy should be one consideration when comparing different classification systems and their accuracy. No extensive work documenting the relationship between the scope of the genre taxonomy, the complexity of the discrimination algorithm and accuracy with the use of benchmark genre taxonomies has been published to our knowledge. In one report, McKay mentions a higher computational complexity in feature selection for the comparison of similar genres. Also, the classification of root genres was higher than that of leaf genres. This agrees with the intuition of increased classification difficulty for specialized genres. The current work does take on the challenge of performing classification in a specialized taxonomy of classical genres.

1.2.3 Other Related Research

We mention some work that has been done that does not directly address the problem of musical genre classification, but that serve as frameworks for generic music analysis tasks.

1.2.3.1 Computing on Music Represented as Strings

Some interesting work has been done in utilizing string representation of music for genre classification (Pérez-Sancho et al., 2005). Works, however, often view strings as a sequence of musical symbols and do not work with techniques directly related to linguistics or natural human language processing. Because of its inherent characteristics, structures in music often require different data models and algorithms than those commonly used in textual applications. In this introduction, we highlight some peculiarities and challenges of computation on symbolic music when compared to natural language text.

Music contains polyphony, the occurrence of multiple notes at the same time. When related to text, polyphony could be compared to multiple sentences that should be read in parallel by different people, all belonging to one logical textual unit. Each instrument or melodic line is not to be interpreted in isolation, but their interactions become an important part of the music perception and ultimate analysis. As an example, consider the

first 26 measures of J. S. Bach's cantata in Figure 1. Note how the text bound to the notes is performed in parallel in the four voices.

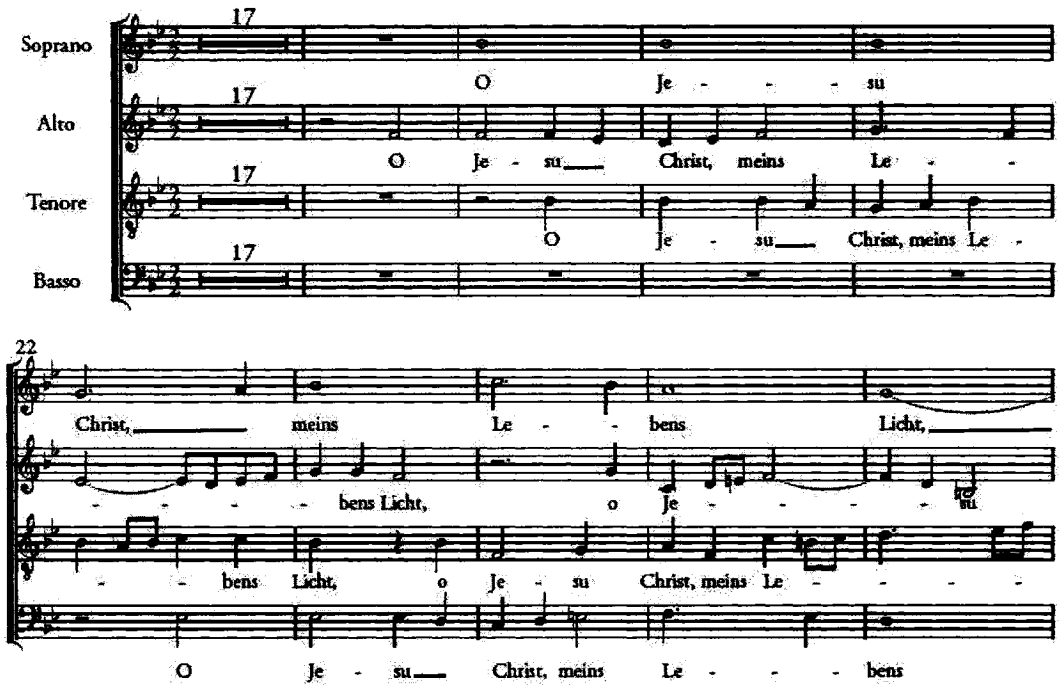


Figure 1: "O Jesu Christ, meins Lebens Licht", J.S. Bach cantata (BWV 118)

Music has the notion of phrases and dialog, sometimes between instruments. For example, consider the beginning of the fourth movement of César Franck's sonata:

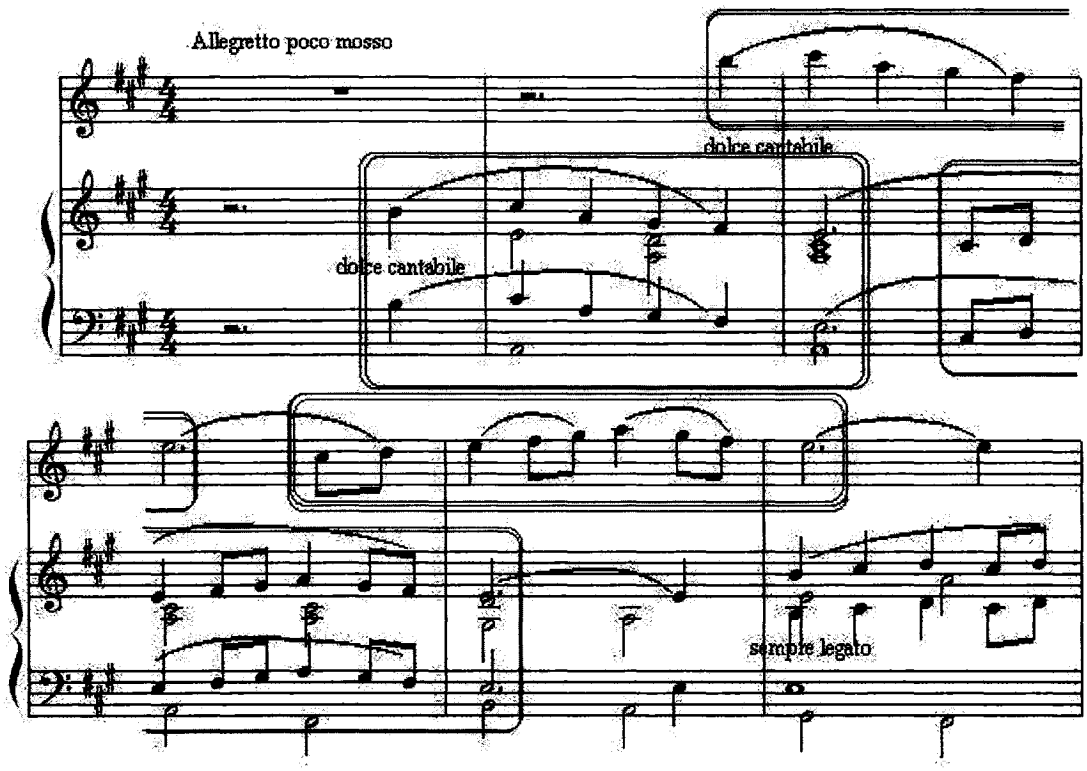


Figure 2: Dialog between piano and violin in César Franck's Sonata in A Major, Movement 4

Not only is the comparison with text classification difficult because of polyphony in the example, but in a more general sense it is still currently difficult to delineate syntax. A well-known work related to music and grammars is Steedman's (Steedman, 1984). Steedman defines a grammar for modeling chord sequences common in jazz and blues. The work, however, is restricted in scope to chord sequences in jazz and blues and does not model other musical objects such as melody and rhythm. Particularly, the idea of phrasing in music can not only be derived from harmony, but also from melody. Longuet-Higgins makes other linguistic comparisons including more musical structures:

> "[...] useful analogies may be drawn between music and natural language. Metrical rhythms resemble syntactic structures in being generated by phrase-structure grammars; as for the pitch relations between notes, the tonal intervals of Western music form a mathematical group generated by the octave, the fifth and the third." (Longuet-Higgins, 1994)

Longuet-Higgins then states that

> "[...] the set of all possible melodies in a particular musical style constitutes a language [...]" (ibidem)

Knopoff and Hutchinson say that "the assumption of a simple functional equivalence between the notations of music and speech is misleading", arguing that not all musical information is adequately represented as discrete values (Knopoff & Hutchinson, 1981).

Modulation in music consists of changing the pitch of a line in a fixed interval. The resulting line contains different notes, but is clearly identifiable with the original line. The original segment and the modulated one can be mapped using a bijection. There is no such operation equivalent in text. The concept of meter is explicit at times in specific types of texts, such as poetry. It is not commonly represented, however, in the notation of most written documents. (For a discussion on rhythmic characteristics of spoken language refer to (Rudziński & Moffa, 1993)). The explicit and formal definition of meter is present, in turn, in the vast majority of staff representations of musical pieces. Meter itself can, more often than not, be easily recognized in the performance.

1.2.3.2 Mathematical Music Theory

In the eighties and nineties, groups of researchers actively projected a discipline now called *Mathematical Music Theory*. It employs strong mathematical formalism for the definition and description of musical objects and their relations. The theory is based on abstract algebra (including modules and vector spaces), category theory, algebraic and combinatorial topology, algebraic geometry, and representation theory among other fields. The most prominent researcher in the area is Dr. Guerino Mazzola, founder of the "Zurich School" of Mathematical Music Theory and author of the referential book in the field, "The Topos of Music" (Mazzola, Göller, & Müller, 2002). Mazzola does address classification in musicology, assuming a rather abstract definition of the term "classification". Prominent European researchers in the area include Dr. Thomas Noll, Carlos Agon, Moreno Andreatta and Gérard Assayag – the last three from the Ircam Music Representation Group in France. In North America, researchers of the field include David Lewin and Elaine Chew.

1.3 Proposed Work

In the current work we will look into some of the musical structures that are capable of conveying genre information. We will then seek a select number of useful features derived from the elementary musical structures found in Music Theory. We aim at a good classification accuracy rate in our tests. Even though a thorough interpretation of the musicological characteristics of genre is out of scope, we intend to set methods hereafter to allow such a task.

We take on the tasks of feature creation, feature evaluation and classification of classical pieces according to their genre. These tasks were implemented by a novel feature evaluation procedure and two standard machine learning algorithms detailed in the following chapters. Figure 3 below shows in a Unified Modeling Language (UML) activity diagram these high level tasks of the proposed work. Feature evaluation, as we shall see, not only affects which features are to be used in training and classification but also serves as a tool in itself for the evaluation of genre characteristics by domain experts. One motivation of this research is for the process, as well as the results, to support a better understanding of the musical characteristics of genres. This is one objective that is not explicitly mentioned nor pursued in previous works.

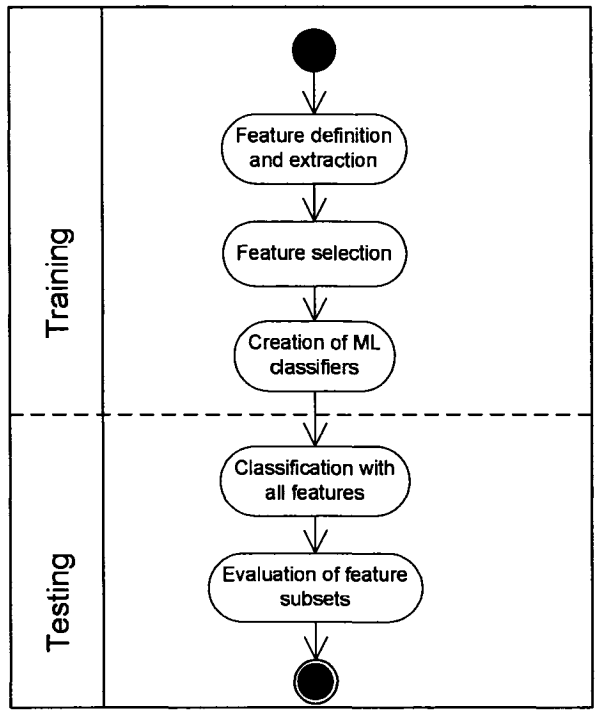


Figure 3: High level UML Activity diagrams: training and testing

The testing steps accept as input a number of unseen music instances and assign genres to them. One measurement of the efficiency of the system is to record how well it can make correct predictions. Given that the system learned some properties of the training data, if the system makes the correct prediction of unseen data we say that the system has generalized and that the quality of this generalization is estimated by the accuracy achieved by the classification algorithm.

1.3.1 The Contribution of the Proposed Work

The main question of this Thesis is: "How can computers find useful features for the classification of symbolically encoded classical music into genres with the help of Machine Learning?" Having summarized state of the art work done on the problem at varying levels of scope and difficulty, this proposed work hopes to contribute to the advance of the research of classical genre classification in a number of ways. First, it defines a procedure for the quantitative measurement of usefulness of feature subsets that

can guide feature creation and increase the classifier's empirical accuracy, with the potential to provide feedback to knowledge experts. The procedure is generic enough for application into other problem domains besides musical genre classification. Generic guidelines for the modification of wrapper methods are included, so that different wrapper methods may help in similar objectives in future works. The proposed work also defines a taxonomy with 14 genres representing of a large period in Music History. It then shows how to represent a few useful features of polyphonic and symbolically encoded classical music for classifications of the classical genres. Specifically, features in the categories of distances in the harmonic möbius strip, rhythmic syncopation and metre and distances in the line of fifths are novel contributions. A set of utilities was created to extract the features, making use of shell scripts and C programs. It also aims at describing experiments with different machine learning algorithms capable of accurately classifying classical musical genres.

Chapter 2 Foundation in Music and Machine Learning

2.1 Introduction

In this chapter we introduce the terminology and concepts used throughout the text. We begin looking at foundational concepts in music, moving on to foundational concepts in Machine Learning.

2.2 Foundational Concepts in Music

Some of the forms and explanations of concepts in this section were borrowed from "The music theory handbook" (Merryman, 1997). The concepts of pitch-class and Schönberg's harmonic möbius strip are not commonly found in introductory bibliography.

2.2.1 General Music Theory

In this section we define basic fundamentals of music, focusing on concepts that will be used later on. The basic unit in Common Western Notation is the *note*, also called pitch. Note names are represented by letters A, B, C, D, E, F, G. Notes may have *accidentals* (alterations) to denote a semitone raise (sharp - ♯) or semitone lowering (flat - ♭). An *octave* designates a *register*, and is a specific height range where the note is situated. At times the octave number is included as a suffix to the note name. An octave contains 12 notes and A4 is standardized today at the 440 Hz frequency[2]. Figure 4 shows A4♯ in the lower left and D♭5 in upper right in Common Western Notation.

---

[2] Throughout history and geographic locations, A4 had other frequencies. Also throughout history, other tunings divided the octave into other parts besides 12, such as 19, 31 and 53 (Straub, 2004).

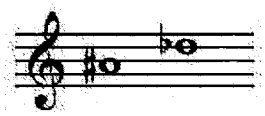


Figure 4: A4♯ and D♭5 in the staff

Although notes in different octaves are different, they sound as equivalent, and have the same base name. This base name is referred to as the *pitch-class*. Figure 5 contains a circular space with the (tonal) label of notes in the exterior and pitch-classes in the interior.

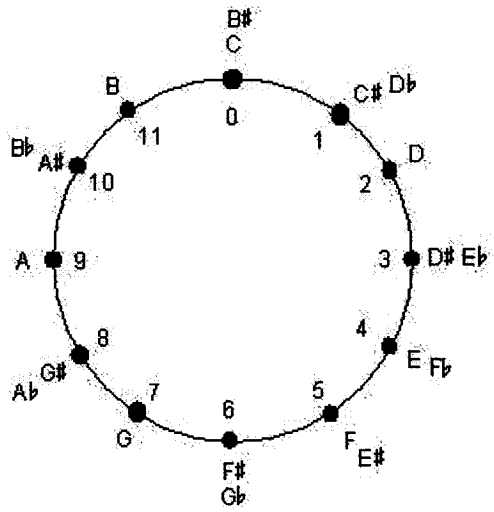


Figure 5: Pitch class space

The *key* of a passage or a piece is a special note that serves as base or reference, along with a *mode*. The *mode* is said to be *major* or *minor*, as explained later. In textual notation, a note letter in upper case denotes major mode, and in lower case minor. The key is an important concept in harmony. Each key is represented by a *key signature*, which identifies the accidentals of the passage. The key signature is indicated in the staff as a set of accidentals, as shown in Figure 6 below.

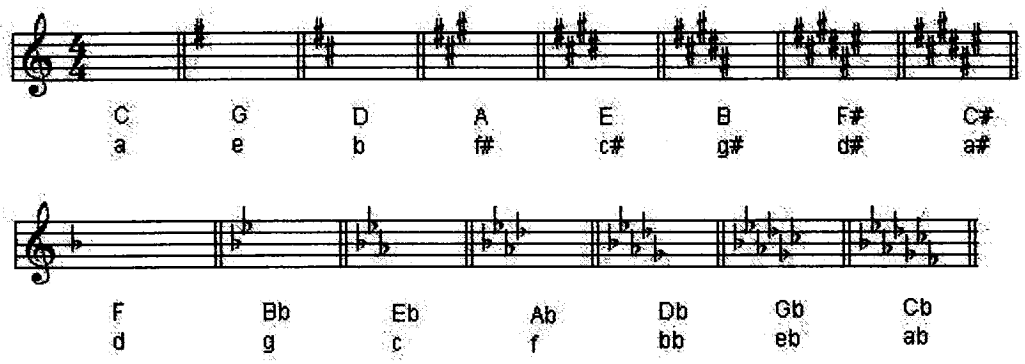


Figure 6: List of key signatures with respective major and minor keys

Major and minor keys that share the same key signature are said to be *relatives*. A *scale* is a sequence of notes, where each note name is in sequence. The white keys in an octave on a piano starting at C form the C-major scale. A scale is defined by the distances between each consecutive note.

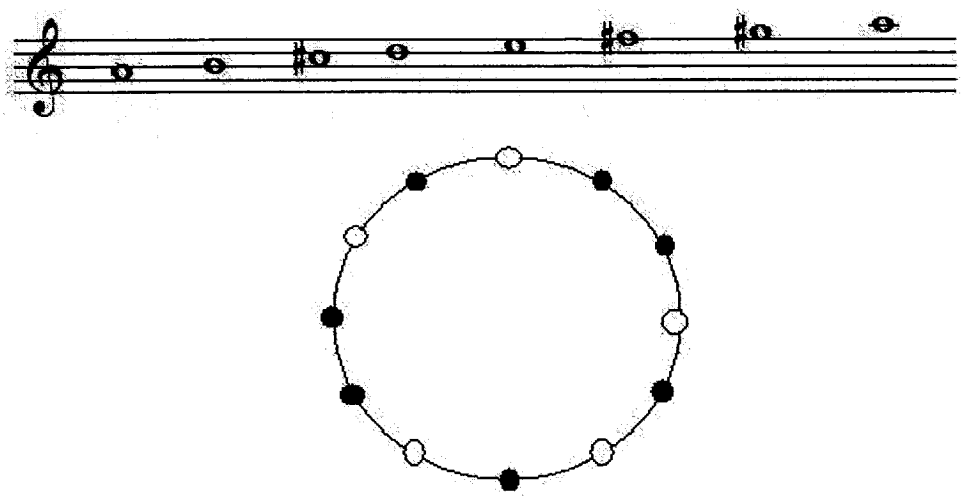


Figure 7: A-major scale in the staff and in the pitch class space

The natural minor scale contains a semitone (half step) between the second and third note, and between the fifth and sixth note.[3]

---

[3] Other minor scales are the harmonic and the melodic minor.

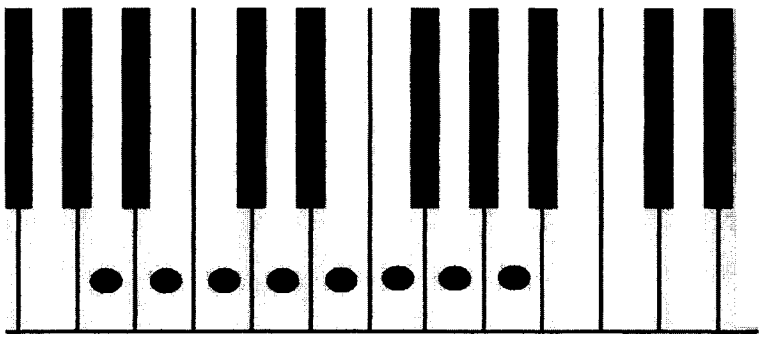


Figure 8: A-minor scale

The notes in a 7-note scale are also referred to as *scale degrees.*

Table 1: Scale degree names and numbers

Pitch in C major	Scale degree number	Scale degree name
C	1	Tonic
D	2	Supertonic
E	3	Mediant
F	4	Subdominant
G	5	Dominant
A	6	Submediant
B	7	Leading tone[4]

The chromatic scale contains 12 notes, with a semitone between each note.

[4] An exception applies in the case of minor scales, where the name is *subtonic* at times

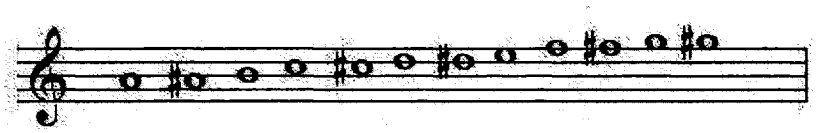


Figure 9: The chromatic A scale

A passage is said to have more or less chromaticism depending on how often chromatic sequences occur or how large they are. Alternatively, a passage is said to have more or less chromaticism depending on how often notes outside common scales appear. An *interval* is a distance between two notes. It may refer to notes occurring at the exact same time or in sequence. Intervals are called *unisons*, *seconds*, *thirds*, *fourths*, *fifths*, *sixths*, *sevenths* and *octaves*, counting over the letter-names. Unisons, fourths, fifths and octaves are said to be perfect intervals. Others are said to be major or minor intervals. Alterations in the distance to any interval are said to be diminished or augmented. A major second interval (M2) has a distance of 2 semitones, a major third (M3) of four semitones, a perfect fifth (P5) of 7 semitones, and an octave of 12 semitones.

Consecutively incrementing an ascending perfect fifth interval, one cycles through all 12 notes of the chromatic scale, arriving back at a pitch of the same pitch class of the starting one. This creates the circle of fifths.

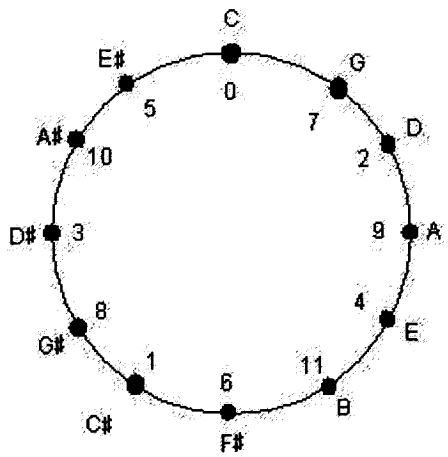


Figure 10: The circle of fifths

The *line of fifths* was proposed by David Temperley (Temperley, 2000). It is similar to the circle of fifths with (tonal) pitch-classes as discrete points, except that it is an infinite line centered on C (see Figure 24). Each note event falls sequentially on the line. The *centre of gravity* is defined as the mean position of all the events of the piece in the line of fifths.

A *chord* is a collection of three notes or more that sound at the same time.[5] A *triad* is a 3-note chord comprised of a root note, a major third up, and a perfect fifth up relative to the root. If the root is the lowest note, the triad (chord) is said to be in root position. Otherwise, the triad (chord) is said to be inverted. If the third interval is major (C-E-G for example), the triad is *major*. If the third is minor (C-E♭-G for example), the triad is *minor*. Other chord *qualities* also exist besides major and minor, depending on the fifth interval. Triads in a passage of a certain key are named after the scale degree of its root. Alternatively, *roman numerals* are used in notation to denote the triad in the given key. By convention, uppercase numerals denote major mode and lowercase minor mode.

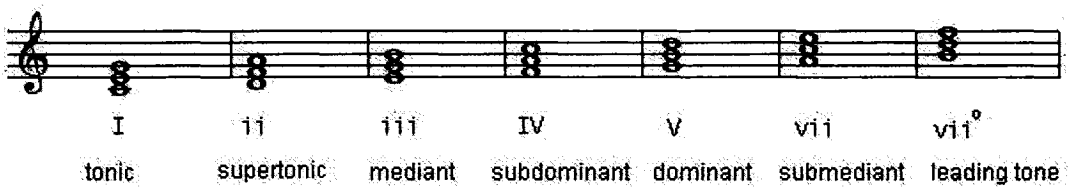


Figure 11: Triads in the key of C major

Schönberg's *harmonic möbius strip* is constructed as follows: each one of the 7 triads of the diatonic scale becomes a point. Draw a connection between each two points whose chords have at least one tone in common. Then, for every triple of chords having at least one tone in common, draw a triangle between the points (see Figure 12) (Mazzola, Göller, & Müller, 2002).

---

[5] This is a simple definition of a chord. Other definitions account for the fact that a chord may be arpeggiated or broken, that is, with notes not sounding exactly at the same time. A more general definition of a chord makes its identification more difficult for the purposes of this discussion.

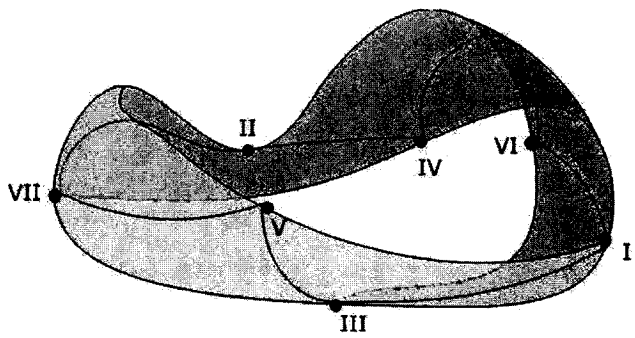


Figure 12: Harmonic möbius strip

A pitch class is a collection of all enharmonic equivalent notes disregarding octave. For example C#4, C#5 and Db3 are all represented by the same pitch class. Pitch classes are represented by numbers from 0 to 11, usually with 0 = C. A pitch class set is a set in the mathematical sense, as defined in Set Theory. Chords can be thus generalized as pitch class sets vertically positioned. When positioned horizontally, pitch class sets can be interpreted as scales. Normal set operations such as union, intersection and complementation are applied to particular sets to derive new sets.

Notes are not unordered events in time. Rather, the natural sense of regularity is reflected in the concepts of beat and meter. All notes and rests have durations specified by their rhythmic value. Rhythmic values are defined as fractions of other rhythmic values. The reference rhythmic value is the whole note. In the same meter, two half notes have the same duration of one whole note; two quarter notes have the same duration of one half note, and so on. Figure 13 shows a rhythmic value hierarchy, with the whole note in the first line, and where all lines have the same duration. Descending one line doubles the number of notes, each one having half of the preceding duration.

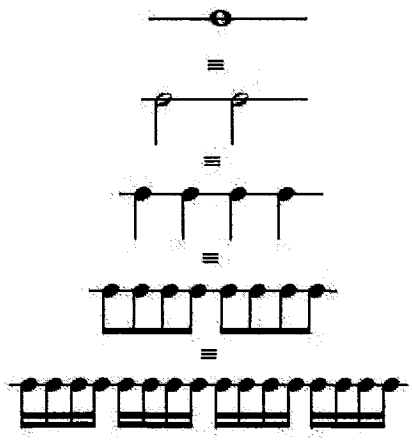


Figure 13: Rhythmic value hierarchy

Beat is a basic pulsation unit. Beats can be accented or unaccented, denoting "strong" and "weak" beats, respectively. *Meter* is the measurement of the period between accented beats. A music passage or a whole piece is organized into *measures*, also called *bars*, according to the meter. All measures of a passage with the same meter contain the same fixed number of beats. The *time signature* of a music passage explicitly declares the number of beats in a measure. The meter can be derived from the time signature, but absolute time cannot, unless a *tempo* is specified. A time signature contains two numbers in a notation similar to a fraction. The bottom number indicates a unit of reference for a beat. The upper number represents how many beats there are in a measure. The time signature is not a fraction, however. A time signature of $\frac{3}{4}$, therefore, defines a span of 3 quarter-notes per measure. A triplet squeezes more subdivisions of a given figure into one beat. Three triplet eights, for example, are equivalent to one quarter note. *Polyrhythms* generalize triplets, allowing the division of note events into some prime number that does not factor the underlying time signature.

Monophony is the phenomenon of one voice performed by one instrument. *Polyphony* is the phenomenon of two or more voices or instruments performing at the same time. Thus, polyphony allows often for more than one note sounding at the same time. The definition accounts for the fact that it is possible for one instrument to perform multiple voices sounding each note sequentially and not exactly at the same time. These occurrences of

polyphony happen less often, though (one bibliographic source states that no algorithm exists currently that is able to automatically separate the voices (Meudic, 2003)).

2.2.2 Genre Definition

Grove Music Online – the online version of the New Grove Dictionary of Music and Musicians (Grove & Sadie, 1980) – contains a definition of genre by Jim Samson:

> "[Genre is] a class, type or category, sanctioned by convention. [...] Genres are based on the principle of repetition. They codify past repetitions, and they invite future repetitions. " (Samson, 2006)

The definition is in fact generic and can be applied to other such branches of the Humanities, such as Literature. Here, the idea of repetition is not specifically concerned with multiple occurrences of particular patterns within one artwork. Rather, genre is the union of various pieces that are grouped according to common characteristics. This still implies the comparability of the various music pieces. Samson continues and comments on the principle of repetition, making the distinction between the music material and its context:

> "The repetition units that define a musical genre can be identified on several levels. In the broadest understanding of the concept, they may extend into the social domain, so that a genre will be dependent for its definition on context, function and community validation and not simply on formal and technical regulation. Thus the repetitions would be located in social, behavioural and even ideological domains as well as in musical materials. A narrower understanding of genre, and a more common usage, separates musical works from the conditions of their production and reception, and identifies genre as a means of ordering, stabilizing and validating the musical materials themselves [...]"

Given that the categorization into genres of music works is also derived from culture and history, it does not follow necessarily that the contents of the music works are insufficient to accurately characterize genre. Compared with branches of Musical studies, only

relatively recently has Computer Science, Math and Statistics provided ways to formally model and compare musical content. These latter sciences may indeed confirm the historical definitions of genre in the future.

Some terms used to denote a particular genre evolved and changed meaning throughout time. For example, Mangsen comments how the term "sonata" has been employed:

> "The rapid development of instrumental music towards the close of the 16th century was accompanied by a plethora of terms which were employed in a confused and often imprecise manner. 'Sonata' was one of them, although it was nearly always applied to something played as opposed to something sung ('cantata')." (Mangsen, Irving, Rink, & Griffiths, 2006)

Note that the criterion "something played" is generic enough to characterize many other music pieces that are not historically classified as sonatas. In fact, sonatas were present in the Baroque period, evolving and changing through the Classical and Romantic periods until the 20th century. Fugue is a genre that originated in medieval times and continued evolving until the 20th century. The new Grove Dictionary of Music and Musicians comments on the difficulty to characterize fugue, to the point of questioning if it is a genre:

> "Despite the prominence of fugue in the history of Western art music and its virtually continuous cultivation in one form or another from the late Middle Ages until today, there exists no widespread agreement among present-day scholars on what its defining characteristics should be. Several factors contribute to this lack of consensus: (1) between the late Middle Ages and the late Baroque a great variety of genre designations – ricercare, canzona, capriccio, fantasia, fugue itself, even motet – came and went in which techniques of imitative counterpoint figured prominently. Thus the history of fugue cannot adequately be accounted for if only pieces called fugue are studied. (2) If all pieces called fugue were collected together and compared, no single common defining characteristic would be discovered beyond that of imitation in the broadest sense. (3) Since the early 19th century genre designations have been defined largely if not exclusively by their formal

> structures. Formal structure, however, is not in the end a defining characteristic of fugue. As a result, there has been prolonged argument about whether fugue is a form at all (and, by extension, whether it is a genre) as well as whether any particular formal model should be considered necessary [...]". (Walker, 2006)

More generically, one comments:

> "[...] while categories like 'genre' or 'style' seem to be used mainly to 'put some order' and reduce the overall entropy in the musical universe [...], sometimes they seem to create even more disorder and confusion" (Fabbri, 1999)

Mazzola overtly criticizes current classification schemes in musicology:

> "We should however stress that classification is highly controversial in musicology. [...] Due to a catastrophical lack of technical tools, traditional musicology has only rarely been able to control the variety of their objects. A disdain of detailed technical work which is psychologically comprehensible must scientifically be blamed for a major scientific retardation even with respect to other humanities such as linguistics." (Mazzola, Göller, & Müller, 2002)

We will acknowledge that there has been some degree of difficulty in finding musical characteristics for a genre definition that are precise and generic at the same time, while relying on the historical division of genres commonly done in musicology.

Bibliography on particular genres is ample. The author, however, did not find one authoritative compilation of classical genres in the bibliography with names of genres separated from those of forms. (In the next section genre and form are compared.) Pachet and Cazaly worked on a novel hierarchical taxonomy beginning with existing metadata from Internet sites and the music industry (Pachet & Cazaly, 2000). The taxonomic scope is admittedly not universal, but includes a very diverse list of 378 genres, including popular genres. The actual taxonomy, though, was not included in the article.

We adopt the definition given by Samson above. As a criterion for choosing genres for the scope of this study, we seek the most descriptive classical genres in Western music. The proposed taxonomy is, therefore, outlined in Figure 14:

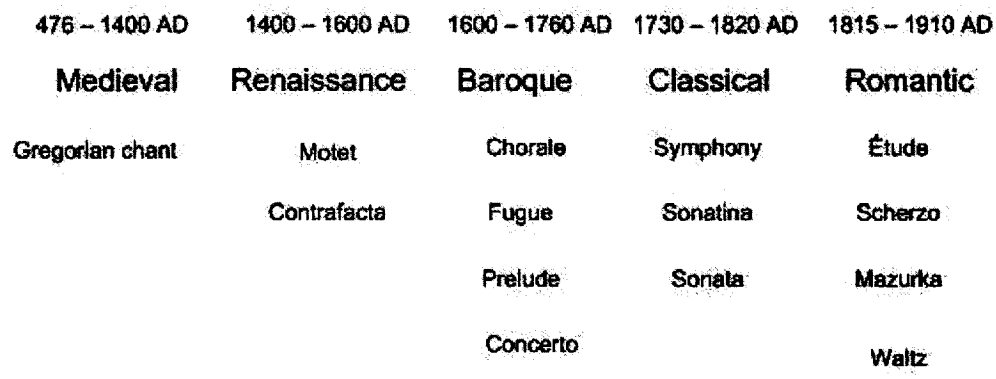


Figure 14: Proposed classical genre taxonomy

In this work, we refer to "classical genres" borrowing the popular meaning of the term "classical". The term generically refers to non-popular works and conveys the idea of music commonly performed by an orchestra, or erudite musicians. Likewise, we attribute this generic meaning to the term "classical music" encompassing such works. Strictly speaking, classical music refers solely to the production of works during the Classical period. We will then employ the term "Classical period" in this strict sense.

Some genres span across multiple periods. One genre may start in one period and become popular or noticeable later. That is the case of the sonata, which had initial works in the Baroque period, but had more expressiveness and popularity during the Classical period enduring change. Scherzo also may be said to be a Classical genre, developing further in the Romantic period. Most of the genres, as a matter of fact, subsist since their beginnings until today with the same name. Similarly, the dates for the periods serve as historical references and do not delineate clear boundaries. We focus not on the arrangement of genres within periods, but on the genres. There are other typical genres, for instance in the Baroque period, that are not included in the taxonomy: the oratorio, the opera and the cantata were noteworthy in the Baroque period. They may enclose multiple, different genres within, such as fugues and chorales in the oratorios (Smither, 1977). One may find also symphonies in introductory movements of oratorios, operas and cantatas (Larue & Wolf, 2006). We do not include them in the taxonomy for this compound characteristic. We do not include modern and post-modern periods, which fall outside what are considered early music (Medieval and Renaissance periods) and the

common practice (Baroque, Classical and Romantic periods). While works of the modern and post-modern periods are considered classical music by some, they are extremely diverse in characteristics. The comparison of modern and post-modern genres among themselves and with those from the Medieval to Romantic periods is a challenging task, beginning with the representation of the musical objects for computation. It is a suggestion for future work.

Regarding the organization of genres, we assume a flat taxonomy: a taxonomy with no hierarchy. The flat taxonomy does not imply that a hierarchical organization of the classical genres is not possible. There are at times classifications that further divide one genre. For instance, sonatas can be classified as church sonatas (sonata da chiesa) and chamber sonatas (sonata da camera), and concerti can be divided into solo concerto and concerto grosso. Usually these terms obtained from further categorization are not labeled "genres", nor is the term "subgenres" commonly applied to them.

2.2.2.1 Genre and Form

Form may be defined as the shape of the collection of all musical elements (Randel, 2003). Thus, when defining form, music theorists compare the structures among different music works trying to find similarities. Historically, from the beginning of the 19th century musicians favoured a definition of genre according to the musical forms (refer to the quotation about the fugue from the New Grove Dictionary of Music and Musicians in section 2.2.2). As a result, forms are well documented in the literature, with more precise terms describing meter, segmentation into sections, instrumentation and other characteristics of the music. There are overlaps of names, for the relationship between forms and genres is close, even though they are considered different types theoretically:

> "There is some overlap between musical form and musical genre. The latter term is more likely to be used when referring to particular styles of music (such as classical music or rock music) as determined by things such as harmonic language, typical rhythms, types of musical instrument used and geographical origin. The phrase musical form is typically used when talking about a particular type or structure within those genres."(Wikipedia contributors, 2006)

As an example, a distinction is made between the sonata genre and the sonata form. The sonata genre may include works from the Baroque period, when the word "sonata" conveyed the idea of a music that is played and not sung. The Sonata form characterizes one single movement and not a complete piece and does not apply to the Baroque period. A Sonata-form movement is said to have three sections labeled exposition, development and recapitulation. Sections differ in the themes present and key modulations. In the beginning of the 20th century, the classification of works according to traditional forms was challenged by the abandonment of tonality by many composers.

2.2.3 Some Characterization of the Chosen Genres

In this section we introduce the genres of our taxonomy. The boundaries between the genres are not apparent at times and their evolution is constant. A thorough musical description is not in order, and we try to compile not historical descriptions, but short ones that can increase the understanding of the underlying domain while including hints of quantitative measures that can be later used in computation.

2.2.3.1 Gregorian chant

The Gregorian chant, or plainchant, was a type of song for to the Roman Catholic liturgy which developed mainly from 800 to 1000 AD. It has been performed since then up to present time. The Vatican II asked in 1963 that the Gregorian chant be given prominence in the Catholic Church liturgy:

> "The Church acknowledges Gregorian chant as specially suited to the Roman liturgy: therefore, other things being equal, it should be given pride of place in liturgical services. " (Sacrosanctum Concilium, 1963)

The Gregorian chant repertory consists mainly of monophonic material, with no instrumental accompaniment. With few exceptions, music follows 8 modes, called Gregorian modes or Church modes, each one with its scale. Each mode also has a final pitch, called a cadence pitch.

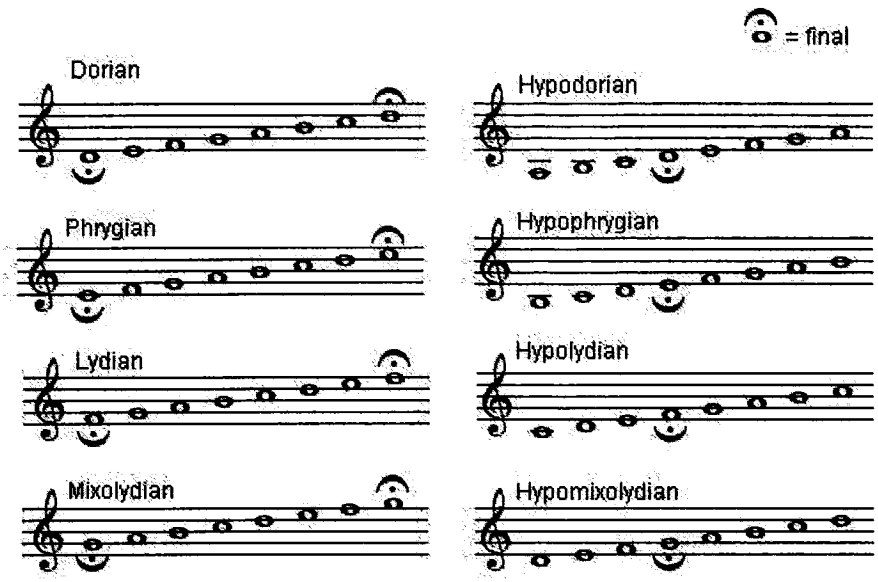


Figure 15: The 8 church modes

In this way, this genre contains more modes than the two common ones introduced during the Renaissance – the Major and Minor modes, which are the most common ones today, even in popular music. The diversity of modality contrasts with the modern harmonic familiarity, so that the modern ear may find the harmony unsettled or unstable. Harmony began to develop later in the period, with groups of boys or castrati singing higher notes, and with the polyphony introduction with women allowed to sing other parts in the liturgy.

2.2.3.2 Motet

Motet was a polyphonic genre that originated in the Middle Ages often in sacred music. The medieval motets contained the idea of repeated rhythmic patterns in all voices, or at least a fundamental voice (tenor). This characteristic changed during the Renaissance. During the Renaissance, sacred texts remained in Latin, contrasting with the vernacular languages employed in the Madrigal genre. The early idea of counterpoint – the melodic interaction of the voices – begins to appear in Motets.

2.2.3.3 Contrafactum

Contrafactum (singular of Contrafacta) refers to vocal music where the text of an original music is substituted without altering substantially the musical part. Often composers would change the secular text with one of religious content, while the inverse happened less frequently. Contrafactum can be a conscious imitation of the original material or simply the quotation of popular melodies in a new arrangement. A challenge, then, resides in that the designation "contrafactum" exists in principle as a modification of some original music and not from derivations of intrinsic characteristics.

2.2.3.4 Chorale

The genre designates a hymn tune of the German Protestant church. Chorale singing was an important part of the Lutheran liturgy from its beginning in the 16th century. It often contains simple melodies intended for a congregation to sing, instead of a professional choir. The melodies often existed previously and were modal instead of tonal. J. S. Bach harmonized many chorales for a four-part choir. The tunes of the chorales served as basic material for the cantata, oratorio, passion and organ chorale genres.

2.2.3.5 Fugue

The quotation in section 2.2.2 mentioned the difficulty in finding common characteristics in all music pieces labeled as fugue beyond that of imitation. It is still worthwhile listing the main characteristics of fugues created during its popularity peak in early 18th century. Commonly, the fugue is an instrumental piece that has a counterpoint where a theme (subject) is gradually introduced in all parts, often transposed. The Latin term fuga is related to fugere (to flee) and to fugare (to chase). After that, its use became much less frequent, but still appearing occasionally at the endings of works in sonata forms or symphonies.

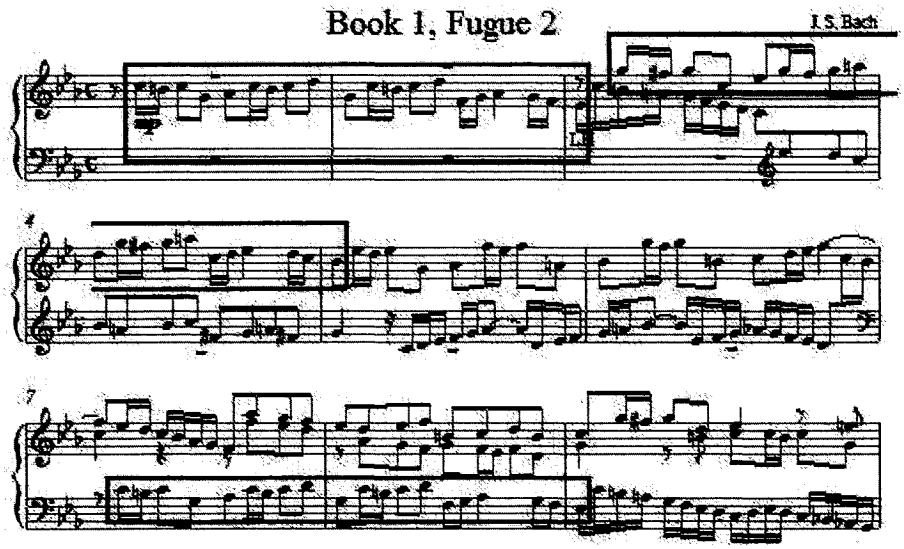


Figure 16: Fugue No. 2 in C minor by J.S. Bach with the identification of subject

2.2.3.6 Prelude

The original notion of the term designates a piece that serves as an introduction to another one. However, since the composition of the 24 preludes by Chopin, preludes can be independent pieces, not necessarily preceding other works. Preludes were often short, for instruments and allowed improvisations. Early preludes were composed for the organ preceding vocal music. Beginning with J.S. Bach, many composers would create preludes in sets, often alternating major and minor keys and cycling through each key in each prelude. For example, each volume of J.S. Bach's Well-Tempered Clavier contains 24 pairs of preludes and fugues. The first pair is in C major, the second in C minor, the third in C# major, the fourth in C# minor, and so on. Beethoven and Chopin – classical and romantic composers respectively – also composed preludes that cycled through keys.

2.2.3.7 Concerto

The concerto is a work for soloist (solo concerto) or a group of soloists (concerto grosso) and a larger orchestra. From its pre-Baroque origins in the 16th century, it encompassed works for instruments and voice. The instrumental concerto then evolved in the Baroque period near the end of the 17th century. In the late Baroque period, a special musical phrase performed by all the orchestra (tutti) called ritornello was repeated from time to

time, while the group of soloists (concertino) played more diverse phrases. The concerto differed from the sonata in that the concerto favoured skilled players, at times hired as concertinos, while the latter would not make such distinction among musicians. In the beginning, most the solos were for violins, whereas gradually compositions were made for most of the string and woodwind instruments. Composers of the classical period introduced forms from the sonata to the concerto. Besides changes in modulations during the exposition, a cadenza is inserted near the end, when the soloist has the opportunity to showcase a difficult technical passage, improvising at times. A concerto is considered one piece and is divided into separate movements played interruptedly.

2.2.3.8 Symphony

Symphony refers to a piece for an entire orchestra. It is subdivided into separate movements. The majority of symphonies of the 18th century are in major keys with key signatures with no more than 4 sharps or 3 flats. They also contained three movements: a fast, a slow and a fast one. In comparison to the Baroque concerto, the early symphony transitioned to changes in melody and phrase length. The symphony also considered the instruments more equal, contrasting with the concerto. By the end of the 18th century, the symphony was a very popular genre. It employed a variety of instruments and usually contained four movements by the 19th century. Gaining popularity, it influenced other genres such as the concerto, the opera, and the cantata.

2.2.3.9 Sonata

The genre is characterized by an instrumental piece of music comprised of three or four movements. It was most often composed for piano solo or the violin and the piano. Sonata is the most important genre of the classical period. A sonata is considered one piece and is divided into separate movements or tempos played interruptedly.

2.2.3.10 Sonatina

Sonatina refers to a short, 'easy' sonata. It gained popularity by the end of the classical period, but did not draw the interest of many romantic composers. Works were composed mainly for the piano solo or piano accompanied by the violin.

2.2.3.11 Étude

Étude ('Study' in English) denotes a short composition conceived for the practice of particular techniques by the musician. Études usually contained some level of difficulty, and their objective oscillated between private practice for technical improvement and public performance. Classical études were often composed for the piano. Chopin and Liszt were composers that created eminent études. A challenge for études consists of finding inherent characteristics across the works beyond technical difficulty that can be derived from their contents, as 'technical difficulty' may be a problematic concept for modeling or computation.

2.2.3.12 Scherzo

This genre is characterized by a $\frac{3}{4}$ meter, rapid tempo and accented rhythm. Frequently, the third movements of symphonies and sonatas were scherzi in compositions of Beethoven. Later authors composed scherzi as independent pieces.

2.2.3.13 Mazurka

The basic mazurka rhythm accents weak beats of a piece in $\frac{3}{4}$ or $\frac{3}{8}$. Often the second beat is accented and less often the third. The accent can be given by emphasis in dynamics or by the rhythmic pattern itself. The genre draws its name from the Polish dance. Chopin, who spent his childhood in Poland, composed a number of well-known mazurkas.

2.2.3.14 Waltz

The musical waltz originated with the dance. It is written in a triple meter, usually with one chord per measure. It became a popular genre in the end of the 19th century and beginning of the 20th century, including works for individual instruments and orchestra.

2.3 Symbolically Encoded Music

Historically, there are three main types of music encoding: sound representation, musical notation and analysis (Selfridge-Field, 1997). Different areas of application will favour different types of encoding formats. For example, the distribution of digital music over the network favours sound representations that allow for compression and high sound fidelity. Composition and publishing favour musical notations. And music pedagogy and musical analysis favour the analytical representations. One encoding may be said to be of one main type, but provide means for applications to perform tasks common to other types of encoding. For example, DARMS (Selfridge-Field, 1997) is an encoding for musical notation, but allows analytical tasks. MIDI is a sound encoding, but allows also analytical tasks. Sound representations focus on how to best represent sound electronically for musical performances on electronic hardware, sequencing, sampling and reproduction. One representation of "raw" sound is the MPEG Audio Layer 3 (MP3).

Encodings for musical notation help musicians and composers to graphically document music. These encodings are concerned with the graphical appearance of music and often allow users to define graphical details that are important to musical execution, editing and composition, such as enharmonic clarity, voice crossing and mixing, ornamentation and lyrics for voice parts. It provides means for layout editting and layout rules. It also provides the necessary flexibility to encode less common notation, such as Gregorian chant, non-Western, Braille and Modern pieces. Examples of file formats include DARMS and the proprietary Finale.

Analytical encodings try to represent the logical content and structure of the music pieces. They try to ease the computational manipulation of music content. Examples include the Humdrum, MusicXML (Good, 2002) and the MuseData (Selfridge-Field, 1997) file formats. MusicXML is an XML language that evolved with Humdrum and MuseData.

MIDI, as we shall see below, is a popular file format with large availability on the Internet that has been historically used for all three purposes.

By symbolic encoding of music we refer to encodings that explicitly contain elementary symbols representing abstract elements in music. More specifically, as it relates to the historical division into three types of encoding, we refer to sound related and analytical encodings that do not represent low level physical measurements of sound.

It is fair to say it is not true that sound-based encoding is the only one capable of capturing music in fidelity and reality. For example, Pierre Boulez is a modern composer that allows the performer to choose the order and repetition of various sections of the piece at the time of performance. Thus, it is possible for the same composition to have many realizations in sound performance being the same in essence. While such compositions can be represented in graphical notation and in symbolic encoding, it is not possible in sound-based encoding. Another example is John Cage's three-movement 4'33" composition, containing 4 minutes and 33 seconds of "silence". The traditional edition of the score contains one page with three movements labelled "tacet", the term that designates that a musician does not play the particular movement. There is no explicit division of time between the three movements, albeit the fact that in the first public execution of the piece the first movement lasted 30", the second 2'23" and the third 1'40". The fact that the piece has three movements cannot be captured in sound-based encoding.

Text can be stored in a Unicode file or a WAV file, the former containing a certain number of bytes per character, the latter containing narrative that are samples of sound over time. A compiler could operate on either format, but most implementations will take Unicode or ASCII as input. A compiler implementation taking raw sound as input would probably transform such representation into another character-based one before operating on the input. Clearly, the problem of speech recognition can be separated from the problems of compilers and Formal Languages. In the same manner, it is not a shortcoming for an analytical application not to operate on raw sound formats.

It is still possible to conceive systems capable of deriving higher-level musical features such as notes, duration, meter, tempo, intensity, segments, bowing, key signature, ornamentation and instruments from sound recordings and such systems could serve not only genre classification, but other algorithms that are able to compute on high level

representation of music, such as segmentation (the division of music into sections) and computer-aided composition.

One can also consider the expressiveness of a certain representation. In the case of a grace note, many criteria exist to determine its execution. The grace note may take a portion of the time from the preceding note, the following note or perceptively fall on the main note. Figure 17 shows the notation for a grace note and the three possible performances.

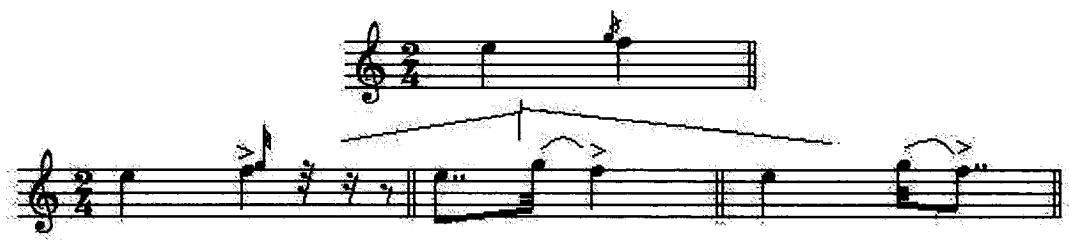


Figure 17: Three possible performances of a passage with grace note

Particularly between the end of the 18th century and beginning of the 19th century different opinions developed as to how to execute the grace note. Any sound encoding of the grace note will realise one of the three interpretations. The semantic of a grace note is therefore lost. Moreover, it is worthwhile commenting that "general rules for an appropriate manner of grace note performance in any given period are impossible to formulate" (Brown, 2006). These are also problems for MIDI, for the notes alone are symbolically represented and stored as they sound, that is, one of the three performances. Even though the notion of a grace note can be represented in MIDI with meta-events, that is not the practice in its current use: the encoding favours the representation of a performance and not of notation or analysis.

2.3.1 File Formats

The main characteristics of the two main file formats used in music analysis follows.

2.3.1.1 MIDI

MIDI is an acronym for Musical Instrument Digital Interface. It originated as a hardware protocol for the exchange of music events between different music devices. The Standard MIDI File (SMF) is a specification of a binary file format. Since its origins MIDI was created for sound applications. Events in MIDI are organized in sequences, with the main events being note events. Note events have five basic attributes: a delta time, note on/note off status, track, note number (pitch) and attack velocity (dynamics).

2.3.1.2 Humdrum

Humdrum (Huron, 2006) is a generic symbolic format conceived for musical analysis. Humdrum encodes different information (notes, harmonic analysis, lyrics, Schenkerian graphs, dance steps, etc.) related to one piece of music. This property allows a more scalable and flexible incorporation of information into the pieces when compared to MIDI. Kern – Humdrum's most common representation – symbolically encodes a variety of primitive musical signifiers found in Common Western Notation, including concert pitch, accidentals, arpeggiations, durations, n-tuplets, meter signatures, tempo and generic ornaments among others. In this sense, Kern's primitive symbols provide richer semantics than MIDI. The Humdrum Toolkit is a framework that allows the manipulation of music pieces encoded in the Humdrum formats. Figure 18 shows the initial measures of an Étude in Hudrum's kern format.

```
!!!COM: Scriabin, Alexander
!!!CDT: 1872///-1915///
!!!OTL: Etude Op. 2, No. 1
!!!OPR: Three Pieces, Op. 2
!!!OMD: Andante
!!!OPS: Op. 2
!!!ONM: No. 1
!!!OCY: Russia
!!!ODT: 1887///
**kern  **kern  **dynam
*staff2 *staff1 *staff1/2
*clefF4 *clefG2 *clefG2
*k[f#c#g#d#]    *k[f#c#g#d#]    *k[f#c#g#d#]
*M3/4   *M3/4   *M3/4
*MM92   *MM92   *MM92
=1-     =1-     =1-
*       *^      *
8CC#/ 8GG#/ 8G#/L       8E/ 8c#/L       8r      p
*       *v      *v      *
*       *^      *
8GG#/ 8G#/      8E/ 8d#/        4E\     .
8GG#/ 8G#/      8c#/ 8e/        .       .
[8GG#/ 8G#/     8c#/ 8f#/       4c#\    .
8CC#/ 8GG#/]    8e/ 8g#/        .       .
8GG#/ 8G#/J     8c#/ 8e/ 8cc#/J 8r      .
=2      =2      =2      =2
8CC#/ 8FF#/ 8C#/ 8F#/L  8.b/L   8B\ 8c#\ 8f#\L  >
8C#/ 8F#/ 8c#/  .       8B\ 8f#\        .
.       16a/Jk  .       .
8C#/ 8F#/ 8c#/  2a/     8A\ 8f#\        .
8C#/ 8F#/ 8c#/  .       8f#\    .       .
8C#/ 8F#/ 8c#/  .       8A\ 8f#\        .
8C#/ 8F#/ 8c#/J .       8A\ 8f#\J       .
=3      =3      =3      =3
*^      *       *       *
8AA/ 8A/L       8C#\    8F#/ 8c#/ 8f#/L 8r      <
8GG#/ 8G#/      4C#\    8A/ 8g#/        4c#\    .
8FF#/ 8F#/      .       8f#/ 8a/        .       .
```

Figure 18: First measures of Étude Op. 2, No. 1 by Alexander Scriabin encoded in Humdrum format

In the kern format, time progresses downward in the file. The header contains editorial information. The file is then structured in columns called *spines*. A spine may encode one voice or instrument, or another aspect of the piece such as dynamics or annotations of harmonic analysis. In each spine, the note duration is represented by a positive integer, with 4 representing quarter-notes, 8 eight-notes and so on. Note names are chosen from {A,B,C,D,E,F,G}, switching case and repeating the characters to indicate the octave. The sharp and flat accidentals are denoted by the '#' and '-' characters, respectively. Any line after the header corresponds to a vertical partition in time of the music. Whenever there are no events happening at a given time partition for one spine but there are for others,

the null token '.' is inserted. Kern's complete file format definition can be found at http://dactyl.som.ohio-state.edu/Humdrum/representations/kern.html.

2.4 Foundational Concepts in Machine Learning

In this section we recall fundamental concepts in Machine Learning, often mentioning concepts contained in Shawe-Taylor and Cristianini's book "Kernel Methods for Pattern Analysis" (Shawe-Taylor & Cristianini, 2004) and Alpaydin's book "Introduction to Machine Learning" (Alpaydin, 2004).

There is a range of problems in Computer Science that are difficult to solve by a direct, explicit computation of an algorithm. Machine Learning is a branch of AI that employs learning methods which explore relationships in sample data to learn and infer solutions. There are many applications where Machine Learning has been successfully applied. Among them there is finding genes in DNA sequences, identifying cancerous cells, handwriting recognition, face recognition and spam filtering. In genre categorization, Xu, Chai and Vercoe, Pérez-Sancho and McKay use Machine Learning.

2.4.1 Patterns

In pattern analysis, learning methods are used to find patterns in data. In the problem of classification, one seeks to predict the value of a special feature in the data as a function of the remaining ones. (In our case, we seek an algorithm that is capable of predicting genre with certain accuracy, with genre being a function of the remaining features.) The consequence is that we can say that a certain dataset contains redundancy: that is, certain features can be derived or predicted from others. Often, it is not always possible to find exact relations in the data that hold all the time. In this case, the algorithm can assert the relations within the data with a certain probability of success. Patterns are the relations within the data that describe redundancy, and can be exact when relations are invariant or statistical when relations hold with a certain probability. By invariant relation we mean a function of the data that always holds for all data.

Predicting one special feature of the data as a function of the remaining features is a common supervised problem in pattern analysis. In that case, training data is represented in the form

$$(\mathbf{x}, y)$$

where y is the target label or class, and $\mathbf{x}$ is the vector with the remaining features.

For this type of problem, the pattern analysis function sought has the form

$$f(\mathbf{x}, y) = l(y, g(\mathbf{x}))$$

where g is the prediction function and l is the loss function. The loss function assigns 0 if and only if the two arguments are the same or a positive number that measures the discrepancy between the correct lable and the predicted lable (that is, a value close to zero when a pattern is detected). For testing, $g(\mathbf{x})$ is used on new data to predict the value of y. In the case of statistical patterns, there is the assumption that the data is generated according to some probability distribution. General statistical pattern for a data source generated independently and identically distributed (i.i.d.) according to a distribution is defined as

$$\mathbb{E}_{\mathbf{D}}\, l(\mathbf{y}, g(\mathbf{x})) \approx 0$$

where $\mathbb{E}$ denotes the expectation of a function according to the underlying distribution $\mathbf{D}$ and the pattern function g is non-negative. Knowledge about the data source is collected from training data generated by the same distribution. Using only this set the pattern analysis algorithm is expected to identify patterns. A pattern analysis algorithm or a machine learning algorithm takes as input a finite set of examples and outputs either no-pattern-found, or a positive pattern function g such that

$$\mathbb{E}\, l(\mathbf{y}, g(\mathbf{x})) \approx 0$$

where the expectation $\mathbb{E}$ is of data generated by the source. The examples are called training data, the pattern function g is said to be the hypothesis returned by the algorithm, the expectation is said to be the generalization error and the pattern analysis algorithm is

also referred to as the learning algorithm. For example, decision trees are hypotheses created by the pattern analysis algorithm. They contain simple decision functions in its internal nodes and output values at its leaves. A classification problem that allows binary output is called binary classification, whereas one that allows a discrete set of outputs is called multiclass classification.

2.4.2 Machine Learning Algorithms

The machine learning algorithm that we seek must have certain properties for it to be efficient. We list three fundamental desired properties: computational efficiency, robustness and statistical stability.

Computational efficiency restricts the class of algorithms to those that can scale with the size of the input. As the size of the input increases, the computational resources required by the algorithm and the time it takes to provide an output should scale in polynomial proportion.

The data that is presented to the learning algorithm may contain noise. In that case, the pattern will not be exact, but statistical. A robust algorithm is able to tolerate some level of noise and not affect its output too much.

Statistical stability is a quality of algorithms that capture true relations of the source and not just some peculiarities of the training data. Statistically stable algorithms will correctly find patterns in unseen data from the same source. When that happens, the algorithm is said to have generalized. The accuracy of such predictions can be measured. A hypothesis overfits the data if it becomes too complex in order to be consistent with the training data. For example, a learning algorithm that is capable of representing any function may give rise to a rote learner; that is, one that simple memorizes the training data without making any inference for unseen data. To overcome the problem of overfitting, the learning machine is biased to consider a subset of all possible patterns in the data. A trade-off between the complexity of the hypothesis and the accuracy is sought. One alternative is to incorporate heuristics into the algorithm with the help of domain knowledge. Two problems with this approach are that not always is domain knowledge available, and that the action tends to customize the algorithm to the specific

problem, making it less general. A second approach fixes the use of standard learning algorithms and recodes the data in an acceptable format. Recoding then amounts to representing the data in an alternative space. This way, one can use standard learning algorithms that produce hypothesis that work in the alternative space and that show the three important efficiency properties mentioned above.

One particular technique works with patterns described by linear functions in a certain feature space. The data recoding exercise, therefore, amounts to mapping the data to the feature space. Linear learning machines are learning machines using hypotheses that form linear combinations of the input data. A special function called kernel is able to implicitly define feature spaces in which the linear learning machines can compute. Kernel methods are, thus, machine learning algorithms that perform pattern analysis in two steps. Firstly, the data is mapped from the original space to a vector space called the feature space. Secondly, a standard linear learning machine learns from the transformed data.

In supervised learning, each input has an associated label used for training. In binary classification, training data has the form

$(\mathbf{x}, y)$

where $y \in \{+1,-1\}$ is the special feature that designates the class of the input $\mathbf{x}$. In the case when $\mathbf{x}$ is a vector, the value of the i^{th} feature of $\mathbf{x}$ is denoted by $[\mathbf{x}]_i$. The variable y is also referred to as label, or target output. Therefore, the training set is denoted by

$$S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)\} \subseteq (X \times Y)^l$$

where l is the number of training examples. In multiclass classification $y \in \{1,2,\ldots,N\}$. The machine learning algorithm estimates a hypothesis

$h : X \rightarrow \{1,2,\ldots,N\}$

that generalizes for unseen data, where X is the input data domain.

2.4.2.1 Kernel Methods

As eluded to before, a kernel employs two steps: a non-linear mapping into a vector space and the classification of data by a linear hypothesis. The idea is illustrated Figure 19 below.

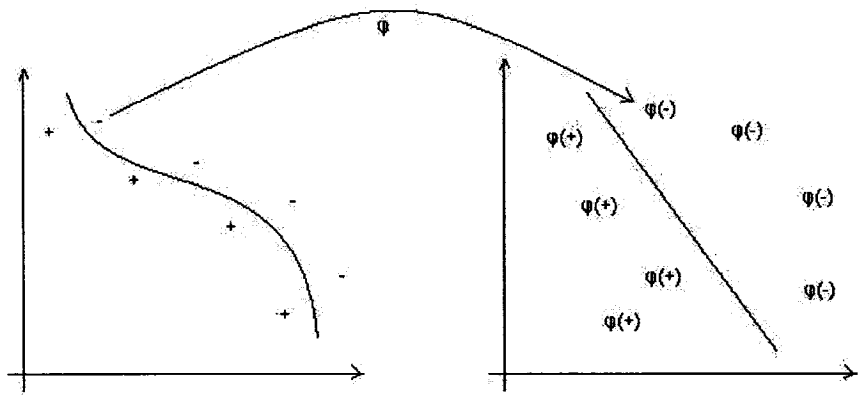


Figure 19: A feature map recodes the data, transforming the original input space into the feature space

In the figure, a two-dimensional input space is transformed via a mapping function φ into a two-dimensional feature space. The input domain X does not have to be a vector space, and can be any countable set. A linear discrimination of the points in the original space is not possible in the former, but is in the latter. Since the function is linear, finding the hypothesis amounts to finding a hyperplane defined by the equation

$$f(\tilde{\mathbf{x}}) = \langle \mathbf{w}, \tilde{\mathbf{x}} \rangle + b = \sum_{i=1}^{n} w_i \tilde{x}_i + b \qquad (1)$$

where $\tilde{\mathbf{x}} = \varphi(\mathbf{x})$. The hypothesis divides the space into two half spaces, and is defined as h(x) = sgn(f(x)), where by convention sgn(0) = 1. Geometrically, the vector w defines a direction perpendicular to the hyperplane, and the value b moves the hyperplane parallel to itself. Rewriting w as a linear combination of the training points $(\tilde{\mathbf{x}}, y)$ yields

$$\mathbf{w} = \sum_{j=1}^{l} \alpha_j y_j \tilde{\mathbf{x}}_j \qquad (2)$$

for scalars α_i. Applying equation 2 into 1, we get

$$f(\tilde{\mathbf{x}}) = \langle \mathbf{w}, \tilde{\mathbf{x}} \rangle + b = \left\langle \sum_{i=1}^{l} \alpha_i y_i \tilde{\mathbf{x}}_i, \tilde{\mathbf{x}} \right\rangle + b = \sum_{i=1}^{l} \alpha_i y_i \langle \tilde{\mathbf{x}}_i, \tilde{\mathbf{x}} \rangle + b. \qquad (3)$$

In the original domain, the hypothesis has the form

$$f(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i y_i \langle \varphi(\mathbf{x}_i), \varphi(\mathbf{x}) \rangle + b \qquad (4)$$

The inner product $\langle \varphi(\mathbf{x}_i), \varphi(\mathbf{x}) \rangle$ can be computed in feature space directly as a function of the original input data without even explicitly computing the mapping φ. In this way, the two steps needed to construct a linear learning machine are merged to produce a non-linear learning machine. A kernel function performs such direct computations.

A kernel is function K: $X \times X \rightarrow \mathbb{R}$, such that for all $\mathbf{x}, \mathbf{z} \in X$

$$K(\mathbf{x}, \mathbf{z}) = \langle \varphi(\mathbf{x}), \varphi(\mathbf{z}) \rangle,$$

where φ is a mapping from X to an (inner product) feature space F.

Kernels can map the data implicitly to the feature space, potentially avoiding computational problems in the feature evaluation. Thus, kernels allow the use of feature spaces with an exponential or even infinite number of dimensions while still not necessarily compromising the above mentioned efficiency requirements. With a kernel function, the decision rule of equation 1 is rewritten as

$$f(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b. \qquad (5)$$

Kernels are modular, in the sense that one can insert a kernel function in any algorithm that represents the data as inner products without change to the algorithm. Conversely, a fixed algorithm can use any qualified kernel. In effect, kernels can be used not only in

classification, but also in other algorithms such as regression, ranking, Principal Component Analysis (PCA) and clustering.

A SVM has a decision function of the form

$$h(\mathbf{x}) = \text{sgn}(\sum_{i=1}^{l} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b) \qquad (6)$$

Finding an optimal hyperplane in feature space is an optimization problem which can be solved using quadractic programming. There are efficient algorithms for the optimization problem which compute in polynomial time.

SVMs are indeed robust learning machines. When searching for linearly separating hyperplanes in feature space, SVMs try to optimize the generalization bounds defined in Generalization Theory (Cristianini & Shawe-Taylor, 2000). Particularly, slack variables in the optimization problem allow margin constraints to be violated by noisy data. Statistical stability of SVMs is guaranteed by different generalization bounds. For example, one bound on the error of an SVM may be a function of the size of the training data, the internal number of support vectors and a confidence level.

2.4.2.2 Bayesian Networks

Consider a feature set A. A Bayesian network B over a feature set A is a directed acyclic graph (DAG) over A and a set of probabilities

$$B_A = \{p(a \mid \pi(a)) \forall a \in A\} \qquad (7)$$

where $\pi(a)$ is the set of parents of feature a in B_A (Bouckaert, 2004). That is, the Bayesian network defines for each vertex the conditional probability given its parents. When feature values are discrete, a table for each vertex contains all the conditional probabilities for the possible values that each parent can assume. The learning altorithms that determine the structure and the probability tables of the Bayesian network may employ different search and scoring criteria. Scoring criteria can be divided into local

scoring, cross-validation and conditional independence tests, while common search methods such as hill climbing, tabu search and simulated annealing can be used.

The determination of the class y of an instance $\mathbf{x}$ is given by the hypothesis

$$h(\mathbf{x}) = \arg\max_y P(y \mid \mathbf{x}) \qquad (8)$$

where $P(y \mid \mathbf{x})$ is estimated as

$$P(y \mid \mathbf{x}) \cong \prod_{a \in A} P(a \mid \pi(a), \mathbf{x}) \qquad (9)$$

2.4.2.3 Decision Trees

Decision trees are a machine learning algorithms that build predictive models by constructing a tree in which each node m contains a test function $f_m(\mathbf{x})$ with discrete outcomes and each arc to a child represents one outcome. A leaf of the tree in the case of classification represents the class assigned to the input instance. When constructing the decision tree, different criteria exist to determine which feature to choose. In a univariate tree, each node contains a test function that considers one feature of the input.

If the features are numerically ordered, such as in the case of continuous values, then the output of the test function is discretized by choosing threshold values, such that for the value $[\mathbf{x}]_i$ of the i^{th} feature we have

$f_m(\mathbf{x}) : [\mathbf{x}]_i \geq w_{m0}$

where w_{m0} is the value of the threshold.

In tree learning algorithms, there may be various trees that correctly classify the training set. One general objective is to seek the smallest tree possible. The problem, however, is NP-complete (Quinlan, 1986). Commonly, a greedy algorithm is employed which seeks the best split possible. Common criteria are based on Information Theory, such as information gain and gain ratio (see section 2.4.3).

The C4.5 algorithm (Quinlan, 1993) is a classical tree learning algorithm that extends on its predecessor ID3 (Quinlan, 1986). Figure 20 shows the pseudocode for the ID3 algorithm. The NodeEntropy is defined in equation 10 and SplitEntropy is the negative summation of the entropies for a certain split, that is

$$-\sum_{j=1}^{n}\frac{N_{a'j}}{N_{a'}}H(a')$$

where a' is a node evaluating feature a, $N_{a'}$ is the number of training instances reaching node a' and $N_{a'j}$ is the number of training instances of $N_{a'}$ taking branch j.

Pruning substitutes a subtree with a labeled leaf. It may be used to reduce the size of the tree with the objectives of minimizing overfitting or the susceptibility to noisy data. Prepruning establishes criteria for stopping tree construction. For instance, if the number of training data available is too few, the tree construction algorithm may stop splitting to avoid a higher generalization error. In postpruning there are no early stop criteria or backtracking. Instead pruning happens at the end of the process. The majority of processes for pruning use an estimation of error for the tree and its subtrees. If the replacement of a certain subtree by a leaf or one of its frequently used branches leads to a lower error rate, then the subtree is pruned.

Input: Training set X, entropy threshold θ
Output: Decision tree

```
GenerateTree(X)
    If NodeEntropy(X)<θ then
        Create leaf labelled by majority class in X
        Return
    i ← SplitFeature(X)
    For each branch of [x]_i do
        Find X_i falling in branch
        GenerateTree(X_i)

SplitFeature(X)
    MinEnt ← MAX
    For all features i = 1,..., d do
        If [x]_i is discrete with n values then
            Split X into X_1,..., X_n by [x]_i
            e ← SplitEntropy(X_1,..., X_n)
            If e<MinEnt then MinEnt ← e; bestf ← i
        Else
            For all possible splits
                Split X into X_1, X_2 on [x]_i
                e ← SplitEntropy(X_1, X_2)
                If e<MinEnt then MinEnt ← e; bestf ← i
    Return bestf
```

Figure 20: Pseudocode of ID3 classification tree construction algorithm

2.4.2.4 Random Forests

Random Forests (Braiman, 2001) is a Machine Learning algorithm that generates sets of decision trees for classification. During testing, the class of an input instance is chosen by the votes of the trees in the forest. Each tree is created with the following algorithm:

1. If the number of instances in the training set is l, sample l cases at random with replacement from the original data. This sample will be the training set for growing the tree.
2. If there are $|A|$ features, a number $m \leq |A|$ is specified such that at each node, m features are selected at random out of the $|A|$ and the best split on these features is used to split the node.
3. Each tree is grown to the largest extent possible without pruning.

Random Forests combine many classifiers aggregating their results. One main variation to the original method is boosting. Boosting considers the accuracy history of trees, changing the weight of their vote accordingly. The original method of counting votes by majority is referred to bagging.

The only two parameters available are the number of features m to consider and the number of trees in the forest.

2.4.3 Feature Selection

Feature selection provides different means for Machine Learning systems to select distinct sets of features with the intent of simplifying the learning process while trying to maximize accuracy. The decrease in the number of features may simplify the problem model and decrease the computation time during training and classification. Feature selection methods are categorized as filters, wrappers or embedded (Guyon & Elisseeff, 2003). Filter methods utilize statistical measurements such as correlation, entropy and information gain to evaluate the merit of each feature without a learning algorithm. Wrapper methods measure the merit of a subset of features with a target learning algorithm. While filter-based methods may provide faster computation than wrapper-based ones, the lack of use of the classifier may produce a feature set that yields suboptimal classification accuracy or performance. Embedded methods are specific to types of learning machines. Feature selection in the latter is an integral part of the learning process and happens during training.

We now present information gain and gain ratio (Gray, 1990), two ranking criteria based on Information Theory commonly used in filter-based feature selection. Later on, information gain and gain ratio will be used in our experiments along with three other ranking criteria.

Let X be the input space. The value of the i^{th} feature of input vector $\mathbf{x}$ is denoted by $[\mathbf{x}]_i$. The *entropy* of a discrete feature a is

$$H(a) = -\sum_{[x]_a \in A_a} P_a([x]_a) \log P_a([x]_a) \qquad (10)$$

where A_a is the finite set of possible values of feature a and P_a is the distribution of a. Intuitively, a more uniform distribution will yield higher entropy than a varied distribution. The entropy of class Y and a discrete feature a is defined as

$$H(Y,a) = -\sum_{y,[x]_a \in A_a} P_{Ya}(y,[x]_a) \log P_{Ya}(y,[x]_a) \qquad (11)$$

where P_{Ya} denotes the joint distribution of (Y,a), A_a is the finite set of possible values of feature a. The conditional entropy of a class given a feature is

$$H(Y \mid a) = H(Y,a) - H(Y) \qquad (12)$$

Define the information gain of class Y with respect to feature a, or average mutual information between Y and a by

$$I(Y,a) = H(Y) - H(Y \mid a) \qquad (13)$$

The information gain has a certain bias towards features that allow multiple values. The bias is compensated in the gain ratio measurement, making the value relative to the feature's entropy. Define the gain ratio of class Y with respect to feature a, or average mutual information between Y and a by

$$R(Y,a) = \frac{H(Y) - H(Y \mid a)}{H(a)} \qquad (14)$$

2.4.4 Feature Transformation and Evaluation

Feature transformation is the process through which new features are created. Two techniques of feature transformation are feature construction and feature extraction. Feature construction creates new features by inference, adding to the original feature set. Feature extraction creates new features applying functions to the original feature set, usually reducing the total number of features. We adopt a definition of Constructive Induction that is slightly different from that of feature construction, allowing the former to be a manual process (Lo & Famili, 1997). Throughout this work we also employ the term "feature extraction", meaning the task of creating features from some input source.

We refer to feature evaluation as the process of quantitatively assessing the features extracted from some input source.

Given a training set S, a machine learning algorithm L, and a feature set A, feature a is incrementally useful to L with respect to A if the accuracy of the hypothesis that L produces using the feature set $\{a\} \cup A$ is better than the accuracy achieved using just the feature set A (Blum & Langley, 1997). Similarly, we extend the notion of feature usefulness to a subset of features. Given a training set S, a machine learning algorithm L, and a fixed feature set A, feature set B is incrementally useful to L with respect to A if the accuracy of the hypothesis that L produces using the feature set $B \cup A$ is better than the accuracy achieved using just the feature set A.

The usefulness of feature set B in relation to feature set $B \cup A$, training set S and machine learning algorithm L is

$$u(B, B \cup A, S, L) = acc(f_L^S[B \cup A]) - acc(f_L^S[(B \cup A) - B]) \qquad (15)$$

where $f_L^S[\Delta]$ is the hypothesis returned by the machine learning algorithm L applied to feature set Δ and $acc(\lambda)$ is the accuracy of hypothesis λ. By definition, B is incrementally useful to L with respect to A if and only if $u(B, B \cup A, S, L) > 0$.

Chapter 3 Feature Extraction

3.1 Introduction

Musical pieces are not readily available for direct computation in their innate formats. In the initial phase of the training process, we create 64 features from the kern files using Humdrum, C programs and shell scripts. Functions from the Humdrum Toolkit were extended and combined for the extraction of features to our needs and we have selected several useful features described below. The Machine Learning algorithm will train and test on data containing these features. In our experimentation, the extraction of the features outputs one XML file holding all the data. XML allows the conversion between text-based file formats through XSLT transformation. XML is also a de facto standard for integration between applications. This way, the feature extraction scripts do not create the output in a format that is specific to the feature ranking and training applications. Should feature ranking or training applications change in the future, no change is needed in the feature extraction scripts.

It is worthwhile noting that all features to be defined here are not specific to classical genres. Also, no one feature tries to single out one or another specific genre. Rather, features were sought that belong to main areas of music theory. In futures research they could well be applied to non-classical genres, for instance.

3.2 Feature Definition

Features are separated into seven categories: (i) distances in the harmonic möbius strip, (ii) distances in the line of fifths, (iii) scale, (iv) rhythmic syncopation and meter, (v) polyphony measurements, (vi) duration and (vii) instrumentation. The definition and further extraction of the features constitute the first step in our experiment.

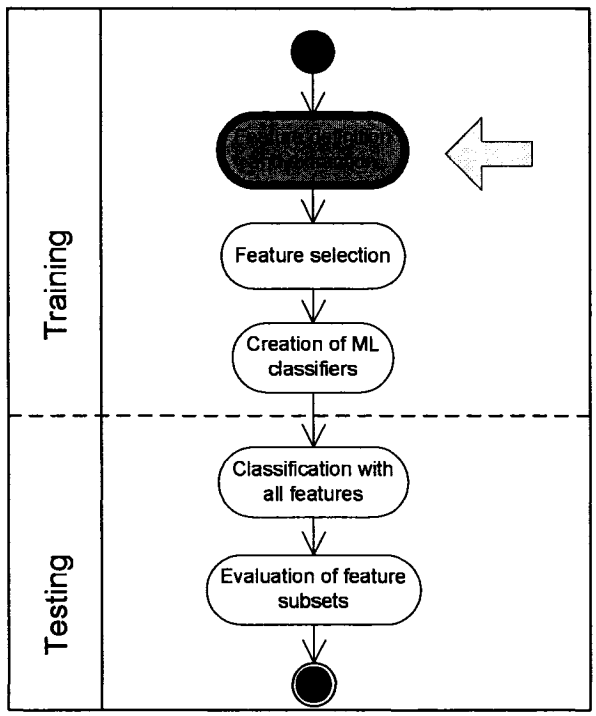


Figure 21: The feature definition and extraction task in the set of experiments

3.2.1 Distances in the Harmonic Möbius Strip

For the extraction of features in this first category, we take Humdrum's sequence of chords in roman-numeral notation as input. The inference of the harmonic content is performed by Humdrum's tsroot command which, in turn, encapsulates Melisma's harmony program (Sleator & Temperley, 2006). Our utility program then models the harmonic möbius strip and performs distance calculations in it.

For each triadic change, the shortest distance in the strip between the two triads is calculated. Three features in the first category are accumulated möbius distance (acc-moeb-dist) throughout all the piece, möbius distance per measure (moeb-dist-me) and möbius distance per change (moeb-dist-chng). The möbius distance per measure is the accumulated möbius distance divided by the number of measures (bars) in the music. The möbius distance per change is the accumulated möbius distance divided by the number of chordal changes. The fixed distance along the strip boundary from the current triad to

triad I is also stored. For example, the distance from II is 2, and the distance from VI is 3. This yields the other three features in this category: accumulated möbius tonic distance (acc-moeb-ton-dist) throughout the whole piece, tonic möbius distance per measure (ton-moeb-dist-me) and tonic möbius distance per change (ton-moeb-dist-chng).

As another example, consider Figure 22 below, containing the first two measures of a chorale. The roman numerals depict the harmony. For simplification, the notation is not including chord inversions or suffixes for notes above the triads[6].

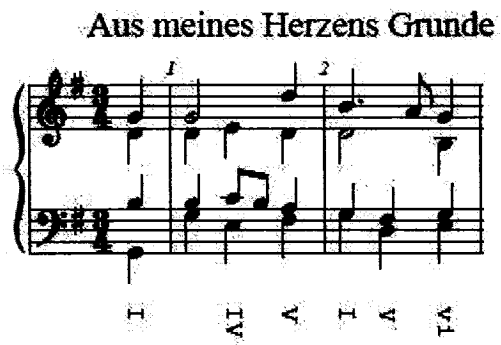


Figure 22: "Aus meines Herzens Grunde" chorale (BWV 269) with harmony in roman numerals

Even though Humdrum's input includes chord inversions, suffixes and chord qualities, the distances in the möbius are calculated only based on the triad. The chord sequence is (I, IV, V, I, V, VI). Considering it a complete piece only for illustration, the accumulated möbius distance is then 1+2+1+1+1+2=8. The möbius distance per measure[7] is 4 and möbius distance per change is 8/5=1.6. The path of the example is illustrated in Figure 23.

---

[6] The notation is not including diminished or augmented qualities either.

[7] The anacrusis is not considered a measure.

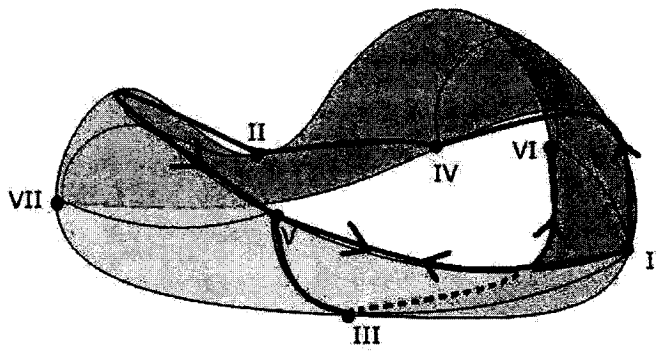


Figure 23: Path of the chord sequence (I, IV, V, I, V, VI) in the harmonic möbius strip

The motivation behind this category of features is to differentiate pieces according to how they follow or respect functional tonality.

3.2.2 *Distances in the Line of Fifths*

The first feature in the second category is the distance from the centre of gravity of the line of fifths to the key (rlofcog[8]).

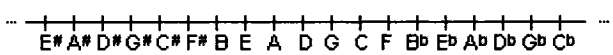


Figure 24: Temperley's line of fifths

Calculating the distance from the centre of gravity to the key turns the first feature invariant to modulation. The second feature is the accumulated deviation from the centre of gravity (rlofcog–adeviation) in the line of fifths measured as a deviation. It is defined as

---

[8] "rlofcog" stands for relative line of fifths centre of gravity

$$rlofcog-adeviation_P = \frac{\sum_{V \in P} \sqrt{\sum_{i \in V} |n_i - d|^2}}{|P|}$$

where the piece P is logically divided over time into partitions, each partition V having a set of notes n sounding at the time. This time division is not an extra computational step; the piece is viewed as a chronological sequence of notes, and the outermost summation iterates in this sequence, while the inner summation follows no particular order.

The motivation is that chromaticism will create a greater dispersion of notes from the centre of gravity depending on the genre.

3.2.3 Scale

The third category of features is based on scales. We begin measuring frequencies of occurrences of the 12 (enharmonic) pitch-classes. We identify the piece's main key utilizing Humdrum's key correlation command does not rely on the written key signature. This provides a higher likelihood of operating with a meaningful key. For example, the Bach chorale "Ach Gott, erhör mein Seufzen und Wehklagen" (BWV 254) has no accidentals in the key signature even though it is in D minor[9]. Pitch-class frequencies relative to the key will allow a fairer comparison among pieces, since a piece can be in any key. The objective is to record the modal profile of the piece, while maintaining it key-invariant. Eventual key modulations throughout the piece do not automatically adjust the pitch-class occurrences count,. Twelve features are labeled rpcfi ($i \in \mathbb{Z}_{12}$)[10]. For example, if a piece is in the key of A (the mode is disregarded), rpcf10 is the frequency of occurrence of pitch G and rpcf2 is that of pitch B.

Scale type (scaletype) is another feature. It can assume the values of "pentatonic", "hexatonic", "heptatonic", "chromatic" or "too-few" when the music has five, six, seven,

---

[9] Statistically, only few pieces show such discrepancies between the written key signature and the perceived key.

[10] "rpcf" stands for relative pitch-class frequency.

more than seven or less than five notes respectively. The scale type and rpcfi have some redundancy. Scale type totals the number of rpcf$_i$s greater than zero.

3.2.4 Rhythmic Syncopation and Meter

Two features measure rhythmic syncopation and three represent meter. The first two features are extracted using Humdrum's synco command:

> "The synco command implements a definition of metric syncopation inspired by the work of Lee and Longuet-Higgins [11] [...]. In brief, metric syncopation may be defined as a moment where an expected metric stress is absent. More specifically, a metrically syncopated moment is defined as occurring when no note-onset happens at a moment whose metric position is more important than that of the most recent note onset. " (Huron, 2006)

The numerical values are calculated as the logarithm of the metric position of the previous onset minus the logarithm of the metric position of the current moment.

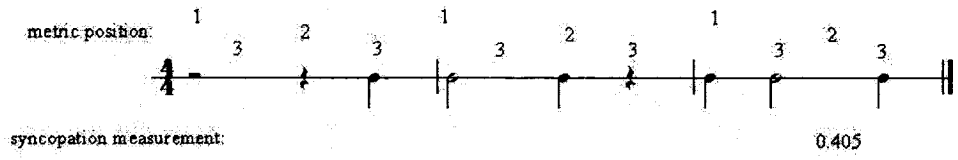


Figure 25: Syncopation measurement

In the example in Figure 25, syncopation happens in the 3rd measure in a moment of metric position 2, where the previous note happens at a moment of metric position 3. The syncopation measurement is ln(3)-ln(2) = 0.405. In the case of polyphony, note events are projected vertically onto one amalgamated line. The syncopation measurement assumes that there is metric throughout the entire piece. Few exceptions can be found: certain fragments of some pieces may be marked as "free form", producing measures with no metric. That is the case of the first movement of Symphony No. 5 in C Minor by

---

[11] Note from author: refer to (Longuet-Higgins & Lee, 1982)

Beethoven. The syncopation measurement assumes also that the music piece can be quantized into discrete positions and that there is a metric position annotation. Other few exceptions apply: the annotation of the metric positions is not sophisticated enough to handle irregular time signatures. Irregular time signatures have a multiple of a prime number greater than 3 in the upper part. Examples include $\frac{5}{4}$, $\frac{7}{8}$, etc. These time signatures are logically decomposed into unit multiples of 2 and 3. For example, $\frac{5}{8} \equiv \frac{3+2}{8}$ and $\frac{5}{8} \equiv \frac{2+3}{8}$. At times, composers make such decompositions explicit in the written notation. With the decomposition it is possible to annotate the metric position taking each subdivisions of the measure. Without the decomposition there is no guarantee that a metric position annotation is correct and the features of syncopation cannot be extracted. In this circumstance, an internal exception is logged, and features in this category will assume a value of zero. Note that the features are still attributed a numerical value, and any data with such exceptional characteristic is still included in the dataset.

The question then arises as to which quantization unit to choose for the metric position annotation. We use a weighted distance between the shortest (s) and longest (l) note:

$$q = \left\lfloor \log_2 l + \frac{3(\log_2 s - \log_2 l)}{4} \right\rfloor$$

The quantization unit is then 2^q. For example, if in a piece the shortest note is a sixteenth and the longest is a half note, then

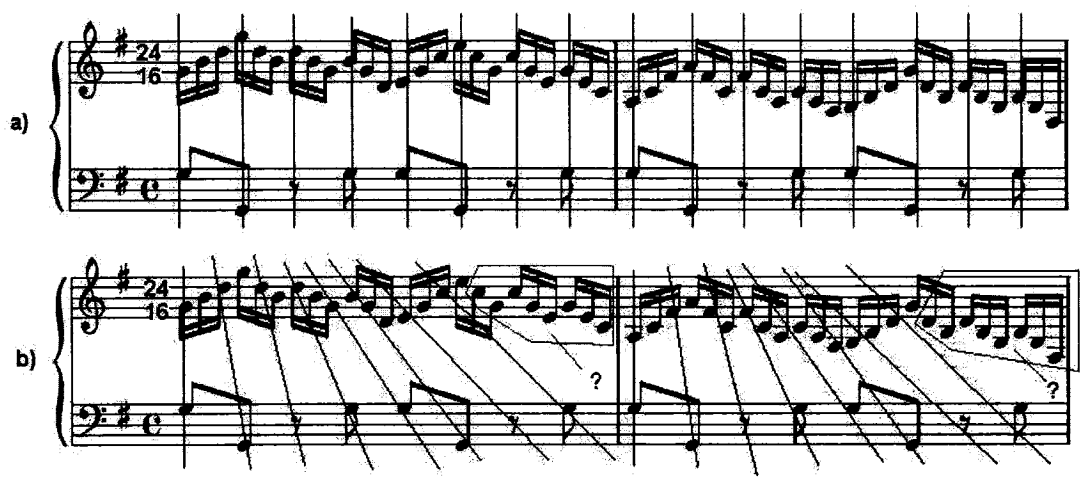
$$q = \left\lfloor \log_2 2 + \frac{3(\log_2 16 - \log_2 2)}{4} \right\rfloor = 3$$

We define the average degree of syncopation (syncop-mean) and the maximum degree of syncopation (syncop-max). The meter of the piece yields three Boolean features depending on the time signatures present: duple (meter-duple), triple (meter-triple) or

irregular (meter-irregular). Note that more than one meter can be present within one piece with time signature changes.

Metric alignment is a property that is necessary for all computations of rhythmic fragments. It is a problem that has been overlooked in all previous works of classification of symbolic music. For the correct comparison of rhythmic events, the meters of the different pieces or even the different voices must be taken into account. Consider the two different metric alignments of Figure 26. The upper system a) shows the correct alignment, with every 3 sixteen-notes in the $\frac{24}{16}$ time signature aligning with one eighth-note in the common time ($\frac{4}{4}$ time signature). The lower system b) incorrectly aligns the rhythmic figures without taking the time signature into consideration. Even if a system utilizes only binary meters, rhythmic figures of triplets, quintuplets and other polyrhythms also force the requirement for alignment.

Figure 26: Correct and incorrect metric alignments of J.S. Bach's Prelude 15 in G major



Our rhythmic feature extraction method performs the alignment and allows for different time signatures in different voices. Metric alignment is necessary not only for computations on music related to classification, but to any application.

3.2.5 Polyphony Measurements

The fifth category includes polyphony measurements. For a given piece, we record the number of simultaneous active notes at each time. This yields three features: maximum polyphonic level (max-polyph), average polyphonic level (avg-polyph) and polyphonic deviation (dev-polyph). The average is calculated taking into consideration each vertical partition:

$$avg - polyph_{\mathrm{P}} = \frac{\sum_{V \in \mathrm{P}} polyph(V)}{|\mathrm{P}|}$$

where the piece P is logically divided over time into partitions, each partition V having polyph(V) notes sounding at the time. The polyphonic deviation is defined as

$$dev - polyph_{\mathrm{P}} = \frac{\sqrt{\sum_{V \in \mathrm{P}} (avg - polyph_{P} - polyph(V))^2}}{|\mathrm{P}|}$$

3.2.6 Duration

The duration of a piece is measured in quarter notes, regardless of the meter. Note that this measures a relative duration of the music and not absolute time, since one quarter note can assume arbitrary absolute duration in music, depending on its tempo and time signature.

3.2.7 Instrumentation

In instrumentation, we capture which instruments or voice designations are in the piece. Each possible instrument is a Boolean feature. The list of instruments is in Table 2.

Table 2: List of instruments

Instrument or voice definition	Feature mnemonic	Instrument or voice definition	Feature mnemonic
Bass	Ibass	Gran Cassa	Igcass
Contralto	Icalto	Oboe	Ioboe
Contrabass	Icbass	Pipe organ	Iorgan
Violoncello	Icello	Piano (pianoforte)	Ipiano
Harpsichord	Icemba	Cymbals	Ipiatt
Clarinet	Iclar	Soprano	Isoprn
Soprano clarinet (in either B♭ or A)	Iclars	Tenor	Itenor
Horn	Icor	Timpani	Itimpa
Bassoon	Ifagot	Triangle	Itrngl
Flute	Iflt	Trumpet	Itromp
Alto flute	Ifltda	Viola	Iviola
Bass flute	Ifltdb	Violin	Ivioln
Soprano recorder	Ifltds	Generic (undesignated) voice	Ivox
Tenor recorder	Ifltdt		

3.3 *Dataset*

The dataset is the collection of files used for feature extraction, training and testing in the system. KernScores (Sapp, 2005) is a library with a large dataset of symbolically encoded music for musical analysis. It contains music pieces encoded mainly in the Humdrum

kern data format. KernScores contained 21,042 files at the time of writing. Some files are copyrighted and have restricted access. Others are excerpts of music for short examples. Others, still, are not categorized into genres or present encoding problems. We considered 943 well-encoded files from KernScores categorized into classical genres and publicly available. All the classical genres can be found in the repository. The classification of genres in KernScores is not accurate at times. It is based on keyword search that matches the header information of the kern files. The header information of kern files does not include genre information in some cases. This casts some doubts over the consistency of the criteria for genre determination in KernScores. While this is sufficient to determine genre most of the time, some exceptions arise. For instance, the third movement of Beethoven's piano sonata No. 10 in G Major (opus 14, No. 2) has scherzo name and form. This paradox is not only related to KernScore's way of assigning genre labels to pieces, but directly related to the evolution of forms and genres. The scherzo began to be introduced to sonatas and symphonies usually as the third movement substituting the minuet. Other composers including Brahms and Chopin began further developed the form, giving them an independent status, from where the generalization of scherzi as genre comes. Because of the keyword search, KernScores allows one file to belong to multiple genres. We followed KernScores' label assignments of genre, and when such conflicts arose, we relied on the distinction made between form and genre for resolution. In the case of the third movement of Beethoven's piano sonata, for instance, we assigned the sonata label to the piece.

Each Humdrum file is considered one input instance, even in the case of Sonatas, Sonatinas, Études and Symphonies that have multiple movements per piece. In KernScores, each movement is encoded as one file. This division into files adds to the complexity of the classification task in the case of pieces made of separate movements, as each instance is treated separately by the system. Movements may have different characteristics from each other, with different instrumentation. We patched 6 files from KernScores with minor encoding corrections. Table 3 summarizes the number of instances per genre.

Table 3: Number of instances of each genre in the dataset

Genre		Instances
1	Chorale	329
2	Concerto	39
3	Contrafactum	24
4	Étude	17
5	Fugue	49
6	Gregorian chant	9
7	Mazurka	53
8	Motet	8
9	Prelude	25
10	Scherzo	5
11	Sonata	319
12	Sonatina	26
13	Symphony	26
14	Waltz	14
Total:		943

There are no particular assumptions made about the music pieces subject to feature extraction other than being capable of encoding in Humdrum's kern file format, which is a flexible format that enables the encoding of virtually all Western music in the historical time span considered.

3.4 Feature Selection

In this section we employ filter-based methods to rank features for subsequent classification experimentation.

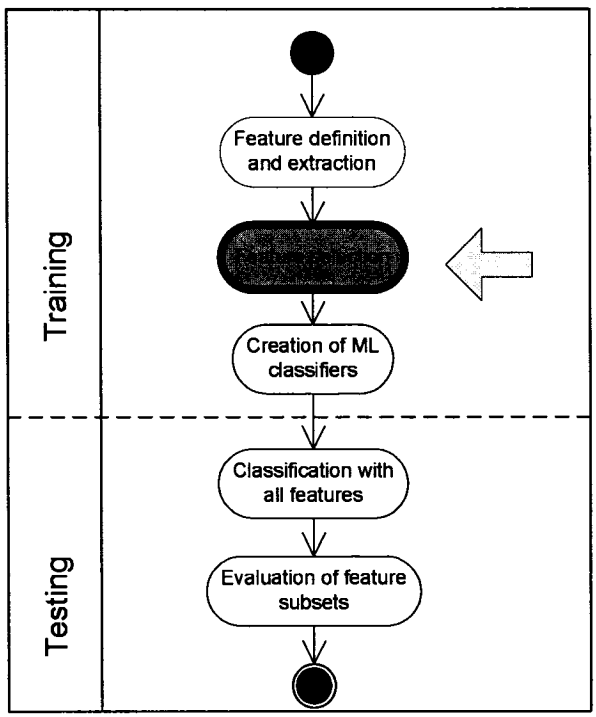


Figure 27: The feature selection task in the set of experiments

Different measurements and algorithms can be used for feature selection. Among filter-based algorithms are RELIEF (Kira & Rendell, 1992), Chi2 (Liu & Setiono, 1995), 1R (Holte, 1993) and Gini (Breiman, 1996). Each may produce a different rank of features. Still, commonalities can be found that may help feature selection and the musicological characterization of genre. Figure 28, for example, shows the gain ratios – one of the feature ranking measurements introduced in section 2.4.4 - for each feature.

In our main feature selection task, we chose to average the normalized scores of five filter-based methods: gain ratio, information gain, Chi2, RELIEF and 1R. Later on, we will remove some of the features that score low in this step.

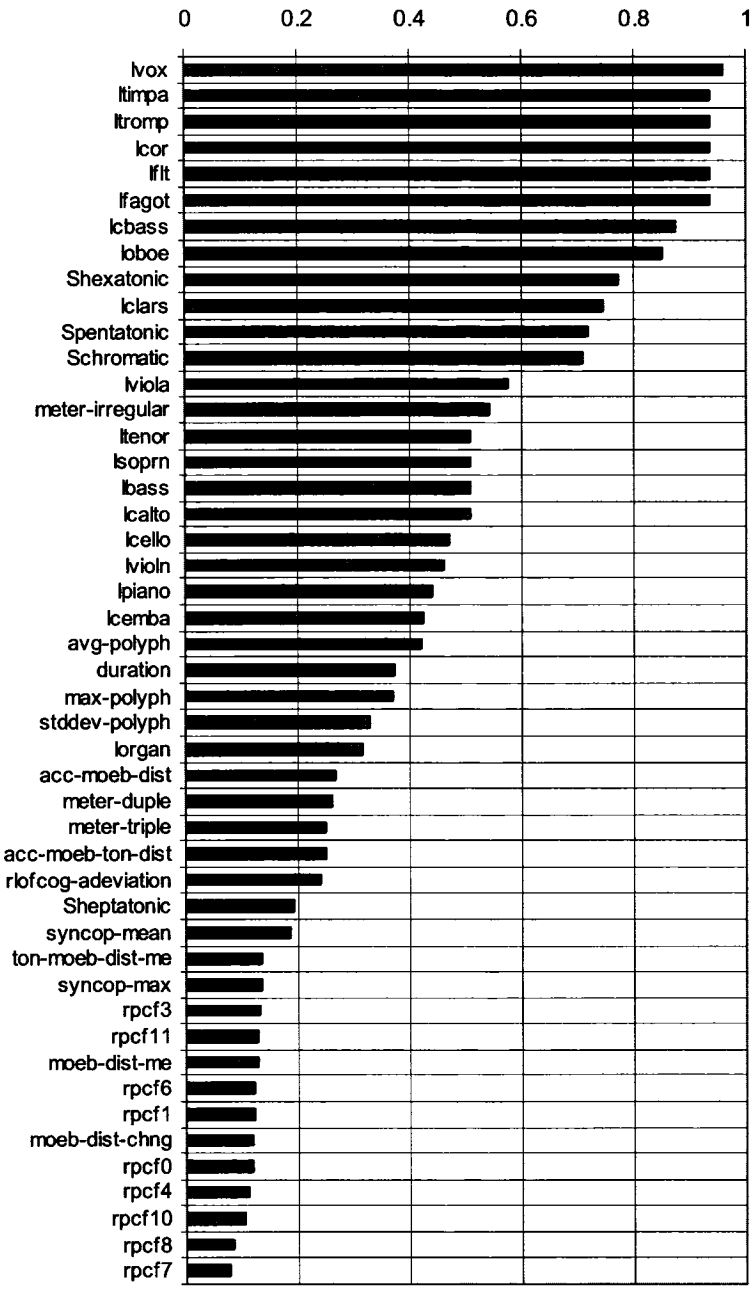


Figure 28: Genre gain ratios with respect to each feature from the dataset

The average of scores of five different filter-based methods was chosen to introduce some diversity in the ranking of the features, considering each method yields a different ranking. Figure 29 shows a summary of the ranking of seven features. The complete

ranking of all features in each method is included in Appendix A. Features related to polyphony and voice rank often high. Similarly, rpcf2 and rlofcog rank often low. Nine features that scored lowest were Igcass, Ipiatt, Itrngl, Ifltda, Ifltdb, Ifltds, Ifltdt, Stoofew and Iclar in decreasing order of score. All nine, except Stoofew, are features that represent musical instruments that appear rarely in the training set. The features that scored highest were avg-polyph, max-polyph, Ivox, stddev-polyph, Ibass, Icalto, Isoprn and Itenor in decreasing order of score. These correspond to the three features of polyphony and the four choir voices plus the generic feature for human voice.

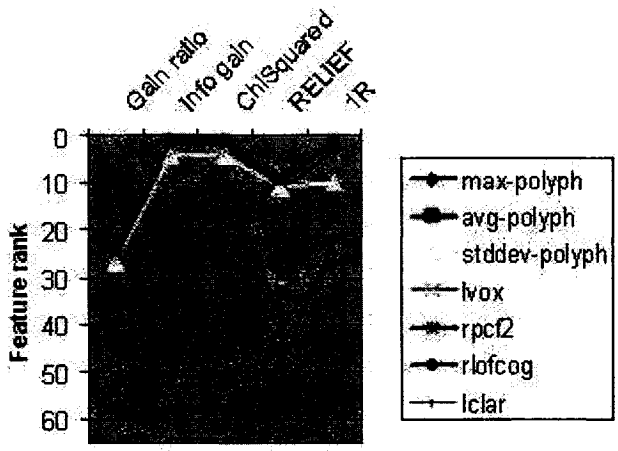


Figure 29: Rank of certain features in different selection methods

3.4.1 Feature Independence

Independence is a property of random variables that asserts that the knowledge of the value of one variable makes the value of another variable neither less nor more probable. In this section we introduce the concepts of independence and correlation with the intent of identifying such relations among features in the particular dataset.

Consider the mean vector $\boldsymbol{\mu}$ of an input vector $\mathbf{x}$, defined as

$$\boldsymbol{\mu} \equiv \varepsilon[\mathbf{x}] = \int \mathbf{x} p(\mathbf{x}) d\mathbf{x} \qquad (16)$$

The covariance matrix $\boldsymbol{\Sigma}$ is given by

$$\boldsymbol{\Sigma} \equiv \varepsilon[(\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^t] = \int (\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^t p(\mathbf{x})d\mathbf{x} \qquad (17)$$

The diagonal elements σ_{ii} (alternatively notated σ_i^2) of $\boldsymbol{\Sigma}$ are the variances of feature i and the off-diagonal elements σ_{ij} are the covariances of features i and j. If i and j are statistically independent then $\sigma_{ij} = 0$. The correlation coefficient

$$\rho_{ij} = \frac{\sigma_{ij}}{\sigma_i \sigma_j} \qquad (18)$$

provides a measurement of how feature i may be linearly approximated as a function of j. $\rho_{ij} = 0$ implies that there is no linear relationship between i and j. $\rho_{ij} = 1$ implies that j is exactly a linear form of i:

$$\rho_{ij} = 1 \Rightarrow [\mathbf{x}]_j = k[\mathbf{x}]_i + \mathbf{d} \qquad (19)$$

Table 4 contains the correlation coefficients above the threshold of 0.95. As the correlation coefficient is symmetric, Table 4 only shows the elements below the diagonal of the correlation coefficients matrix. We point to the exact linear approximation that exists between the four chorale voices – soprano, alto, tenor and bass (Isprn, Icalto, Itenor and Ibass features, respectively). We also highlight the high covariance between the mobiusdist-acc and acc-moeb-ton-dist. This is an interesting property that asserts that statistically the measurement of the shortest distance from consecutive triads can be closely approximated by a linear function of the distance of a triad to triad I along the boundary of the harmonic möbious strip.

Table 4: Correlation coefficients of highly correlated features

	lbass	lcalto	lcello	lcor	lfagot	lflt	lfltda	lfltdb	lfltds	lgcass	loboe	lpiatt	lso-prn		ltim-pa	mo-biusdi st-acc
lcalto	1.00															
lcor																
lfagot				0.99												
lflt				0.98	0.98											
lfltdb							1.00									
lfltds							1.00	1.00								
lfltdt							1.00	1.00	1.00							
loboe				0.98	0.98	0.97										
lpiatt										1.00						
lsoprn	1.00	1.00														
ltenor	1.00	1.00											1.00			
ltimpa				1.00	0.99	0.98					0.98					
ltrng										1.00		1.00				
ltromp				1.00	0.99	0.98					0.98				1.00	
acc-moeb-ton-dist																0.96

3.5 A Feature Usefulness Procedure

In an attempt to quantitatively measure the usefulness of musical features described above, we define the simple FEATUROMETRE procedure.

The approach is similar to that of wrapper methods in the sense that the hypothesis is wrapped within the algorithm. The main difference, however, is in their objectives: while wrapped methods are used in feature selection and output a subset of features, FEATUROMETRE outputs the usefulness of a partition of the feature set for its evaluation. Indeed, wrapper methods rely on the accuracy of the hypothesis to rank features. The accuracy, however, is consumed internally and often weighted with other factors for the assessment of feature subsets and to stop the computation. Another difference is that wrapper methods often employ their own search strategy to iterate through different feature subsets, whereas FEATUROMETRE takes a predetermined feature set partition as input. This gives the user of the algorithm the choice of which

subsets to investigate. The procedure allows a domain expert to quantitatively assess the feature set and gain more knowledge of the data.

Input: feature set *A*, collection *B* of subsets of *A*, machine learning algorithm *L*, training set *S*

Output: base accuracy for feature set *A*, accuracies for *B*, usefulness of *B*

begin

$\alpha = acc(f_L^S[A])$

foreach subset B_i in *B* do

append $\beta = acc(f_L^S[A - B_i])$ to L_1

append $\alpha - \beta$ to L_2

endforeach

return (α, L_1, L_2)

end

Figure 30: The FEATUROMETRE procedure

In general terms, it is conceivable for certain wrapper methods to change to provide the user with measurements of usefulness or scores of feature subsets besides the chosen subset. To that aim, the wrapper method would need to allow two modifications: to output accuracies of the learning machine at different stages and to provide the user with greater control over the feature subset search. Possible options would include a search criterion defined by the user outside the wrapper or the definition of constraints on the feature set to be respected by the wrapper.

In our experiments, the FEATUROMETRE procedure was implemented using a combination of shell scripts and XSLT. The XSLT transformations were applied to the XML representation of the data, incrementally suppressing feature sets before passing the data to the machine learning algorithms.

Chapter 4 Music Genre Classification

4.1 Introduction

Training takes place after the data is extracted. The end of the training task should produce a hypothesis capable of testing unseen data. For training and testing we conduct two experiments using three different SVMs, Bayesian networks, the C4.5 tree learning algorithm and random forests. The first experiment searches for good parameters for the classifiers and performs tests on the complete dataset: that is, the dataset with all attributes. It performs a stratified, nested 10-fold cross validation, detailed below. The second experiment removes features from the dataset trying to improve the classification performance and measure the influence of features in some classifiers. With the exception of one multiclass SVM algorithm, all other machine learning algorithms ran on the WEKA package (WEKA, 2006). The SVMLight (Joachims, 2004) multiclass implementation was tested in the first experiment. The empirical accuracies are reported, assuming a zero-one loss function. The empirical accuracy is the number of instances correctly classified divided by the total number of instances in the dataset. That is,

$$\frac{\left|\{h(x) : h(x) = y\}\right|}{n}$$

4.2 Experiment: Training and Testing with All Features

The first classification experiment is divided in two parts. First, we estimate good parameters for the hypotheses. Second, we train and test subsets of the dataset. Each one of the parts trains a machine learning algorithm and tests its induced classifier.

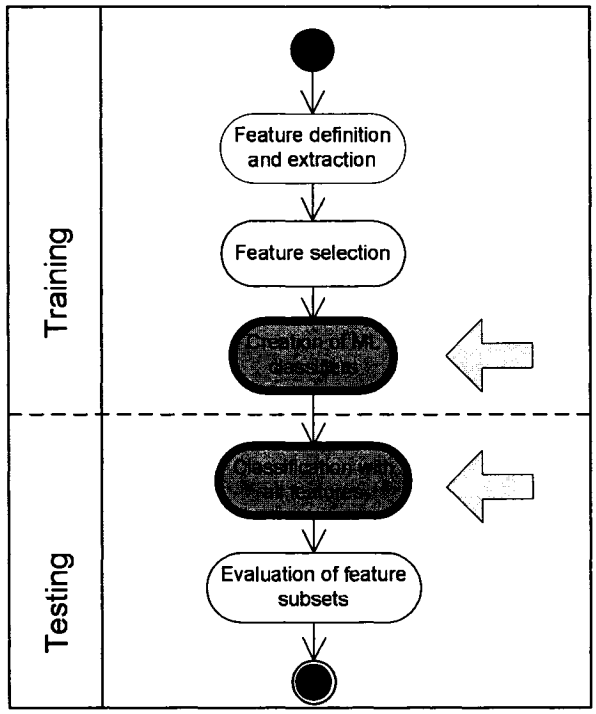


Figure 31: Training and testing tasks with all features in the set of experiments

The motivation for separating the data into training, parametric testing and testing folds is for the training of parametric values not to bias the final accuracy of the hypotheses.

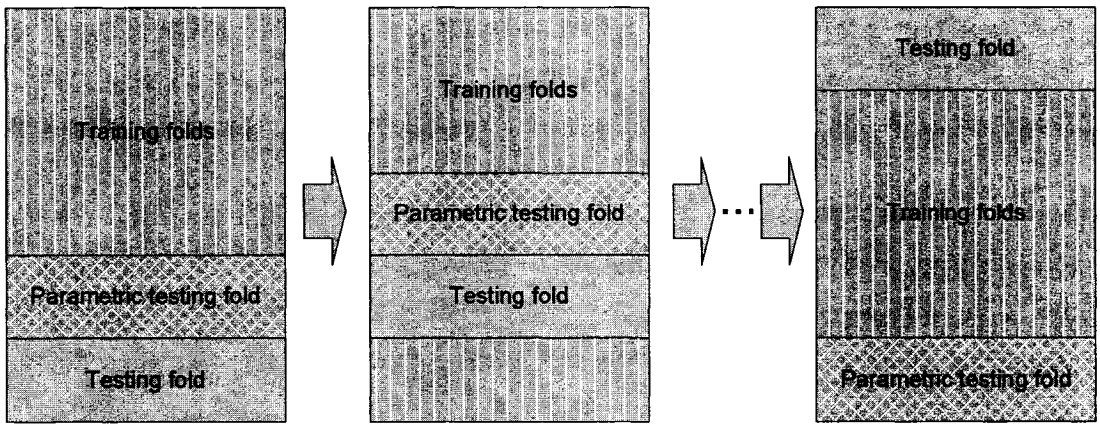


Figure 32: Nested n-fold cross-validation

The estimation of the parameters is performed in a simple way. A few parameters are selected along with sets of arbitrarily chosen values. Then all their combinations are tried

in the inner 10-fold cross-validation step. Five runs are performed and the accuracy of the hyphotheses are recorded and averaged. The parameters that perform best are then fixed. The learning machine then undergoes another round of 10-fold cross-validation – called outer cross-validation, retraining on what were the training and testing fold of the inner cross-validation and testing on unseen data with the fixed parameters. The outer cross-validation process repeats 10 times and the accuracies are averaged. In both inner and outer cross-validation process, the data is shuffled just before it is divided into folds. The cross-validations are said to be stratified, for when shuffling the data they try to maintain the same original distribution of the genres within each fold.

4.2.1 Parametric Search

4.2.1.1 Support Vector Machines

The Sequential Minimal Optimization algorithm (Platt, 1999) is used to train the SVMs, optimizing constraints analytically. Two particular kernel functions were tested. The polynomial kernel is defined as

$$K_p(\tilde{\mathbf{x}}, \tilde{\mathbf{z}}) = \langle \tilde{\mathbf{x}}, \tilde{\mathbf{z}} \rangle^d$$

Besides varying d in the parametric search, we experiment with two data preprocessing options: normalization and standardization. In normalization we set

$$\tilde{\mathbf{x}} = \hat{\mathbf{x}} = \frac{\mathbf{x}}{\|\mathbf{x}\|}$$

In standardization, we define

$$[\tilde{\mathbf{x}}]_i = [\mathbf{x}']_i = \frac{[\mathbf{x}]_i - m_i}{\sigma_i}$$

where m_i is the mean value of attribute i, and σ_i the standard deviation of i. Note that standardization in this case centers the feature value around the mean and makes it proportional to its own standard deviation. The transformation of one feature is not,

however, a function of the covariances with other features. The process of whitening can sometimes be used in conjunction with kernel-based methods such as SVMs in the case of low-dimensional data (Shawe-Taylor & Cristianini, 2004). The whitening process finds an eigen-decomposition to alter the data by scaling the feature space adjusting the size of the eigenvalues, effectively creating a feature space where the data distribution is spherically symmetric. The technique is more commonly applied to the kernel function image, however, instead of the kernel domain as a preprocessing step.

Table 5 summarizes the results of the 10-fold inner cross-validation for different parametric values. We fix $d = 2$ and the normalization transformation for the next tests with the polynomial kernel, which are the parametric choices that yield the maximum accuracy.

Table 5: Accuracy results for parametric search for polynomial kernel in stratified 10-fold cross-validation

With data normalization				
	$d = 1$	$\boldsymbol{d = 2}$	$d = 5$	$d = 8$
Average accuracy	77.86 %	**82.67 %**	79.75 %	78.83 %
Standard deviation of accuracy	0.48	0.75	0.47	0.88
With data standardization				
	$d = 1$	$d = 2$	$d = 5$	$d = 8$
Average accuracy	81.78 %	80.17 %	73.64 %	61.17 %
Standard deviation of accuracy	0.75	0.71	2.56	1.69

Recall the modularity property of kernel methods that allows different kernel functions mentioned in section 2.4.2.1. As long as the kernel properties of symmetry, finitely positive semi-definitiveness and continuity are satisfied, we can substitute the different kernels in the SVM hypotheses. Therefore, we define the Gaussian kernel as

$$K_g(\hat{\mathbf{x}},\hat{\mathbf{z}}) = \exp(-\frac{\|\hat{\mathbf{x}}-\hat{\mathbf{z}}\|^2}{2\sigma^2})$$

The kernel allows the definition of the σ parameter.

Table 6: Accuracy results for parametric search for Gaussian kernel in stratified 10-fold inner cross-validation

	σ = 0.4	σ = 0.625	**σ = 0.75**	σ = 0.9
Average accuracy	75.86 %	79.80 %	**80.25 %**	79.04 %
Standard deviation of accuracy	0.63	0.68	0.35	1.01

We fix σ = 0.75 for the next tests with the Gaussian kernel. The parameters d and σ control the capacity of the classifier. As the parameter d increases, it further allows the hypothesis to more closely fit the target data, characterizing overfitting. Similarly, small values of σ tend to produce the same effect of large values of p. Conversely, small values of d and large values of σ will underfit the data. The choice of d = 2 for the polynomial kernel denotes a hypothesis described by a polynomial a degree 2 in feature space. Therefore, the hypothesis does not tend to overfit, since the value of d is small. For the parametric choices for the Gaussian kernel, the higher value of σ also indicates that the hypothesis is not overfitting the data.

The Gaussian kernel was implemented by the author and integrated into the WEKA package. Binary SVM classifiers were trained and later combined to form multiclass classifiers. Classification by pairwise coupling is the algorithm used during the testing phase that allows the combination of binary classifiers (Hastie & Tibshirani, 1998). It extends the idea of votes by a set of binary classifiers. Pairwise coupling takes advantage of the fact that most classifiers not only select one class, but also provides a probability. Pairwise coupling then combines the votes and the individual probabilities into a joint probability estimate for all classes. This is a common approach in the fusion technique (Kuncheva, 2004). In fusion, classifiers are combined, where each classifier computes on the complete feature set. Another similar technique is classifier selection, where each classifier knows well part of the feature space.

The multiclass SVM with polynomial kernel implemented by SVMLight was not included in the parametric tests. This implementation will be tested, however, in the second part of this experiment with the polynomial degree d fixed at this stage and normalization.

4.2.1.2 Bayesian Network

For the Bayesian Network classifier, we experiment with two search methods: repeated hill climber and simulated annealing. Repeated hill climber is a greedy graph search algorithm that chooses the best node in the neighbourhood of the current node according to the evaluation function. Simulated annealing is an optimization algorithm that searches a new node in the neighbourhood of the current solution varying the probability of choice according to a decaying variable called the temperature. In our tests we fixed the starting temperature at 10 and and the decaying factor at 0.999.

Table 7: Accuracy results for parametric search for the Bayesian Network classifier in stratified 10-fold inner cross-validation

	Repeated Hill Climber	**Simulated Annealing**
Average accuracy	82.97 %	**83.47 %**
Standard deviation of accuracy	0.71	1.09

As results in Table 7 show, we choose the simulated annealing search method for the Bayesian network classifier.

4.2.1.3 Random Forest

In the parametric search for random forests, we vary the number k of features to be considered in random selection and the number i of trees in the forest. Table 8 shows that the accuracy does not varied much across the different parameter combinations.

Table 8: Accuracy results for parametric search for the Random Forest classifier in stratified 10-fold inner cross-validation

	k=7		**k=16**	
	i=22	i=35	i=22	**i=35**
Average accuracy	83.12 %	83.45 %	83.21 %	**83.82 %**
Standard deviation of accuracy	0.58	0.58	0.49	0.76

4.2.2 Results and Analysis

In the outer cross-validation, as explained above, retraining and testing with the fixed parameters 10 times for each 10-fold cross-validation yields the results outlined in Figure 33. Two machine learning algorithms were added in this second part of the experiment: the multiclass SVM with polynomial kernel and the C4.5.

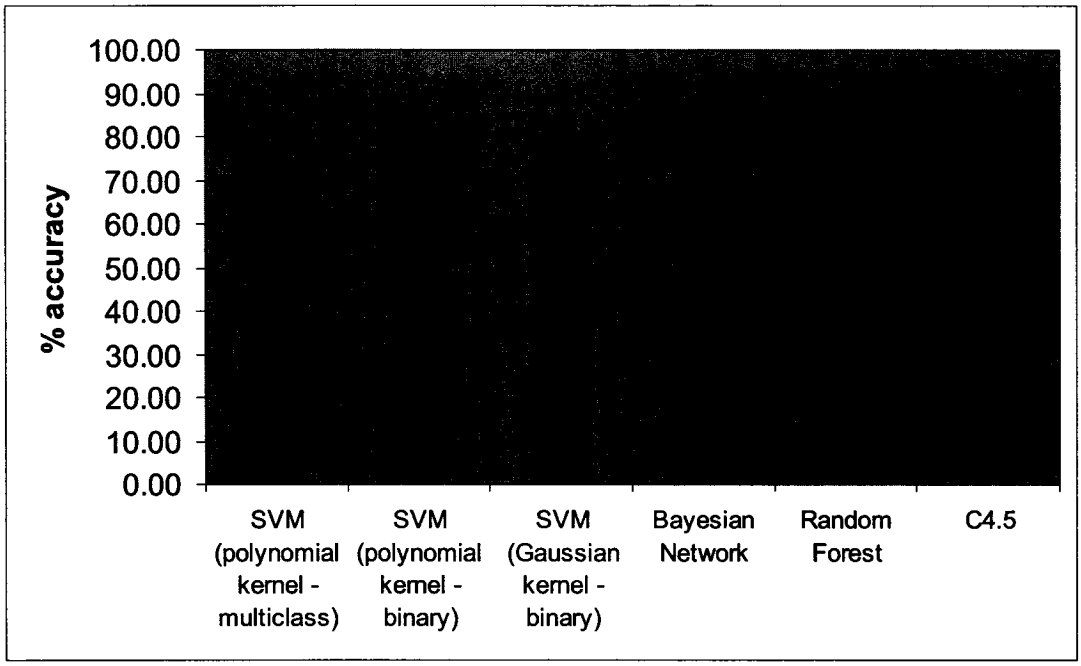


Figure 33: Accuracy results of stratified 10-fold outer cross-validation

The C4.5 algorithm was tested only with outer cross-validation, as no parametric search was performed for it. The multiclass SVM with polynomial kernel was trained and tested using the same fixed parameters of the binary SVM with polynomial kernel, that is with $d = 2$ and normalization. The implementation of the multiclass SVM with polynomial kernel (the SVMLight implementation) does not combine binary classifiers like the previous two SVMs (WEKA implementation). Instead, its hypothesis formulation allows for the direct classification into multiple classes (Crammer & Singer, 2001). The multiclass SVM with polynomial kernel does not use Sequential Minimal Optimization either during training. Consequently, training time for the multiclass SVM implementation is much longer than that for binary SVMs. Despite these differences, the implementation is the one that performs worst not only when comparing it to the binary SVMs but to other classifiers as well.

The C4.5 algorithm produced a pruned tree that performs well. However, the fixed parameters contain a tendency to overfit, producing a decision tree with 58 leaves and 115 nodes. Some of the leaves were created to classify 2 or 3 examples of the training set. Another test tried the C4.5 algorithm with different pruning rules: the minimum number

of classified training examples in a leave was set to 5, and the general heuristics was changed from the classical C4.5 prune scoring to error reduction. The idea was to improve the generalization capacity of the generated decision tree. The later test carried out with the same cross-validation process of all other algorithms achieved 79.21 % empirical accuracy. The size of the tree decreased considerably when compared with the first tree. The second tree had 19 leaves and 37 nodes overall. The small decrease of 2.12 % in accuracy points to an interesting tradeoff between empirical accuracy and generalization favouring the smaller decision tree. In the first induced decision tree, nodes in the first three levels included the Ivox, meter-irregular, max-polyph, Icor and Ibass features. With the exception of meter-irregular, these are features that ranked high in the previous normalized score of the feature evaluation test. The second decision tree assigned features Ivox, meter-duple, Iviola, Icor and duration to its nodes in the first three levels, which are features that did not score as high collectively as the features in the first decision tree.

With empirical accuracies ranging from 74.66 % to 84.10 %, it can be said that overall the classifiers perform well. Note that the binary SVM classifier with Gaussian kernel performed consistently worst than the binary SVM with polynomial kernel. That was the case in both parametric search and testing phases. Besides having a poorer performance than its binary counterparts, the multiclass SVM implementation also took on average considerably longer time to train. That is because the Sequential Minimal Optimization method allowed a faster training of the binary classifiers.

The Bayesian Network classifier with simulated annealing is the one that yields the highest empirical accuracy: 84.10 %. It is interesting to note that this is a higher accuracy than the one obtained in the inner cross-validation step during parametric search, which was 83.47 %. That is probably because of the larger training data available for the outer cross-validation, which considers the validation fold of the inner cross-validation step as training data. Other classifiers such as the binary SVMs and Random Forests performed with less accuracy in the outer cross-validation test than in the inner cross-validation step.

The induced Bayesian network classifier formed a DAG that in fact had a tree structure of two levels. The first level consisted of a simple structure: the root node which is the

genre and the second level contained all features. In other words, the machine learning algorithm did not create a hierarchy of the features in the Bayesian network classifiers. That is a restriction of the chosen software implementation and not of the algorithm itself. The produced tree makes the Bayesian network classifier in practice equivalent to a naïve Bayes classifier. A DAG with length greater than 1 could reveal dependencies between the features. In that case, the topological sort of the DAG could be another source of information to the musical domain expert for a better understanding of the relations among features and between features and genres.

The confusion matrix in Table 9 details the accuracy of classification for each genre. To aid its interpretation, we introduce the error coefficient e_{ij} which measures the proportion between instances incorrectly classified and the total number of instances for a pair of genres y_i and y_j:

$$e_{ij} = \frac{c_{ij} + c_{ji}}{c_{ii} + c_{jj}}$$

for $c_{ii} + c_{jj} \neq 0$ and defined as 0 otherwise. The variable c_{ij} corresponds to entry with indices i and j in the confusion matrix. The error coefficient e_{ij} allows us to identify pairs of genres that are often misclassified as one another.

Table 9: Confusion matrix as a result of classification with the Bayesian network classifier

Classified as ▶	Chorale	Sym-phony	Étude	Fugue	Prelude	Contra-factum	Sonata	Mazur-ka	Motet	Sona-tina	Waltze	Con-certo	Grego-rian chant	Scherzo
True class ▼														
Chorale	317	0	0	0	4	0	8	0	0	0	0	0	0	0
Symphony	0	24	0	0	0	0	1	1	0	0	0	0	0	0
Étude	0	0	6	1	1	0	6	3	0	0	0	0	0	0
Fugue	0	0	0	40	0	0	7	1	0	1	0	0	0	0
Prelude	4	0	1	0	10	0	4	4	0	0	1	1	0	0
Contrafactum	0	0	0	0	0	24	0	0	0	0	0	0	0	0
Sonata	4	0	14	6	3	0	255	10	0	20	3	4	0	0
Mazurka	0	0	0	0	0	0	2	50	0	0	1	0	0	0
Motet	0	0	0	1	0	0	2	0	5	0	0	0	0	0
Sonatina	0	0	0	0	0	0	5	0	0	21	0	0	0	0
Waltze	0	0	0	0	0	0	1	9	0	0	4	0	0	0
Concerto	0	0	0	0	1	0	11	1	0	0	0	26	0	0
Gregorian chant	0	0	0	0	0	0	0	0	0	0	0	0	9	0
Scherzo	0	0	0	1	0	0	0	1	0	0	0	1	0	2

The pairs of genres with highest error coefficients are Mazurka and Waltze (0.185), Étude and Prelude (0.125), and Sonata and Sonatina (0.090). When analyzing dance genres defined by the rhythm, Zamacois comments that only the Waltze could be mixed with the Mazurka within the dances that subsist until the present, because their meter and movement are more or less similar (Zamacois, 1960). Meter for both genres is usually triple. Zamacois in fact classifies both Mazurka and Waltze as genres defined not by form but by movement or rhythm. One complication arises from the fact that tempo or any other measurement of rhythm that captures absolute time is not present in the feature set.

Mazurkas tend to be not only faster than Waltzes but also more syncopated. Features in the category of rhythmic syncopation and meter do not indicate to convey enough information for the discrimination of the two genres. Even though the error coefficient for Étude and Prelude is relatively high, the absolute number of works misclassified is low (2). Sonata and Sonatina are two genres that in principle are only distinguishable by their duration. The distinction can more easily be done in binary classifiers such as the particular SVM implementations previously tested. In fact, the error coefficient for Sonata and Sonatina with the binary SVM with Gaussian kernel is 0.078, which is lower than the 0.090 produced by the Bayesian classifier.

Note that the Bayesian network machine learning algorithm uses the prior probabilities of the class labels in its calculation of the posterior probability. The posterior probability estimates the likelihood that the true genre of input x is y. That is, a high number of training instances belonging to class y increases the chance of the choice of y by the Bayesian network classifier. Hence, for the particular dataset used, there is a bias in the Bayesian network classifier towards the classification of genres including mainly Sonatas and Chorales, which are the genres with the largest number of training instances. The class prior probabilities appear in the C4.5 and random forest classifiers in the search for the features in the decision trees when the information gain is calculated. That is, influence of the prior class probabilities on the choice of the genre is less direct in the case of C4.5 and random forest than in the case of Bayesian networks. SVMs do not use this prior information, but still achieve similar empirical accuracy.

Table 10: Confusion matrix as a result of classification with the random forest classifier

Classified as ▶	Chorale	Sym-phony	Étude	Fugue	Prelude	Contra-factum	Sonata	Mazur-ka	Motet	Sona-tina	Waltze	Con-certo	Grego-rian chant	Scherzo
True class ▼														
Chorale	322	0	0	0	2	0	5	0	0	0	0	0	0	0
Symphony	1	24	0	0	1	0	0	0	0	0	0	0	0	0
Étude	0	0	1	0	0	0	14	2	0	0	0	0	0	0
Fugue	0	0	0	41	0	0	8	0	0	0	0	0	0	0
Prelude	6	0	1	2	3	0	8	4	0	0	0	1	0	0
Contrafactum	0	0	0	0	0	24	0	0	0	0	0	0	0	0
Sonata	5	0	1	3	0	0	287	6	0	16	1	0	0	0
Mazurka	0	0	0	0	0	0	10	43	0	0	0	0	0	0
Motet	0	0	0	0	0	2	3	0	3	0	0	0	0	0
Sonatina	0	0	0	0	0	0	22	0	0	4	0	0	0	0
Waltze	0	0	0	0	0	0	5	7	0	0	2	0	0	0
Concerto	0	0	0	0	0	0	18	0	0	0	0	21	0	0
Gregorian chant	0	0	0	0	0	0	0	0	0	0	0	0	9	0
Scherzo	0	0	0	0	0	0	1	1	0	0	0	1	0	2

The random forest classifier produced the second best empirical accuracy. The confusion matrix for the classification tests with the random forest classifier is included in

Table 10. The pairs of genres with highest error coefficient are again Waltze and Mazurka, Sonata and Sonatina and Predule and Étude. Comparing with the results achieved by Bayesian networks, in random forests the error coefficient for Waltze and Mazurka decreased to 0.155. The coefficient for Prelude and Étude increased to 0.25 and that for Sonata and Sonatina increased to 0.13.

We can group the results of classification per genre period, verifying with which accuracy the classifiers are assigning genre labels within the same period. This amounts to rearranging the confusion matrix as a square block matrix, where each block becomes a new cell with the sum of the elements.

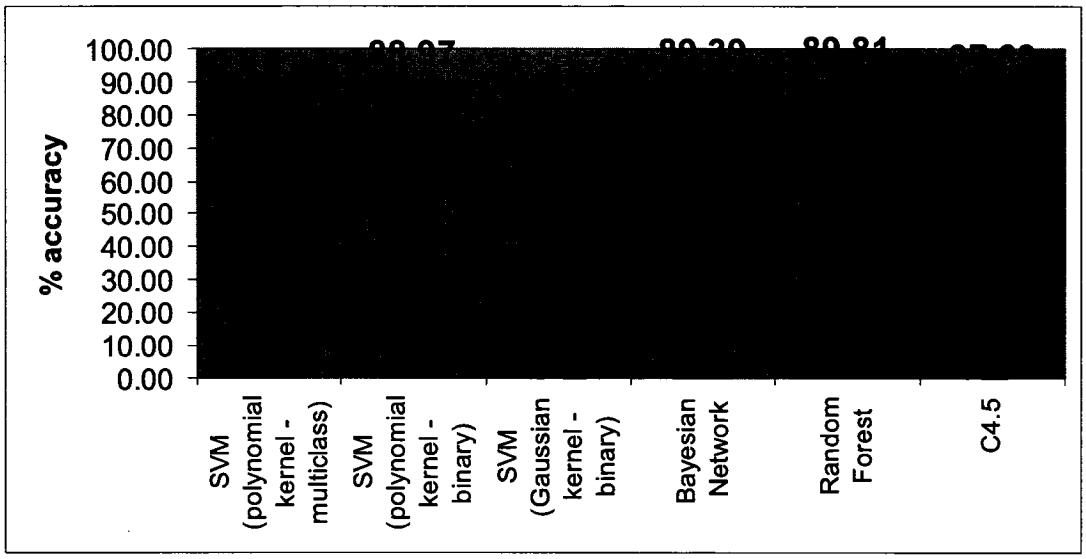


Figure 34: Accuracy results of stratified 10-fold outer cross validation grouped by period

Note that even though the Bayesian network classifier performed best in genre classification, in the rearrangement of the confusion matrix grouping genres by periods the random forest classifier yielded the highest empirical accuracy rate: 89.81 %. The rearrangement in fact forgives genre misclassifications within the same period and cannot decrease the new accuracy rate. The interpretation is that the features and machine learning algorithm still learn underlying characteristics applicable to genres of the same period. From the musicological point of view it is more interesting to have a

classification within the period than to have some other that has an even distribution of genre labeling errors among all genres. Table 11 and Table 12 list the rearrangement of the confusion matrices by period for the Bayesian network and random forest classifiers.

Table 11: Confusion matrix by period as a result of classification with the Bayesian network classifier

Classified as ▶	Medieval	Renaissance	Baroque	Classical	Romantic
True class ▼					
Medieval	9	0	0	0	0
Renaissance	0	29	1	2	0
Baroque	0	0	403	31	8
Classical	0	0	17	326	28
Romantic	0	0	4	9	76

Table 12: Confusion matrix by period as a result of classification with the random forest classifier

Classified as ▶	Medieval	Renaissance	Baroque	Classical	Romantic
True class ▼					
Medieval	9	0	0	0	0
Renaissance	0	29	0	3	0
Baroque	0	0	398	39	5
Classical	0	0	10	353	8
Romantic	0	0	1	30	58

Indeed, when the classifiers infer incorrectly the genre of a piece, they still tend to indicate another genre within the same period of the correct genre. Note that the matrix rearrangements are only interpretations of the original confusion matrices and not results

of hierarchical classifications, since the periodic information was not represented in the training data in any experiment. Differently than other hierarchical genre taxonomies included in other works, period is not considered a genre. The highest error coefficient for the periodic confusion matrices of both Bayesian network and random forest classifiers is for the Classical and Romantic periods with the value of 0.092. Certain composers including Beethoven produced works that are classified into both periods. For instance, our dataset has Classical sonatas and symphonies and Romantic scherzi by Beethoven. Therefore, certain musical characteristics of the composer likely influenced these misclassifications.

4.3 Experiment: Evaluating Feature Subsets

Removing certain features from the dataset, we try to find better classification accuracy as well as verify the influence of features and groups of features upon genre determination.

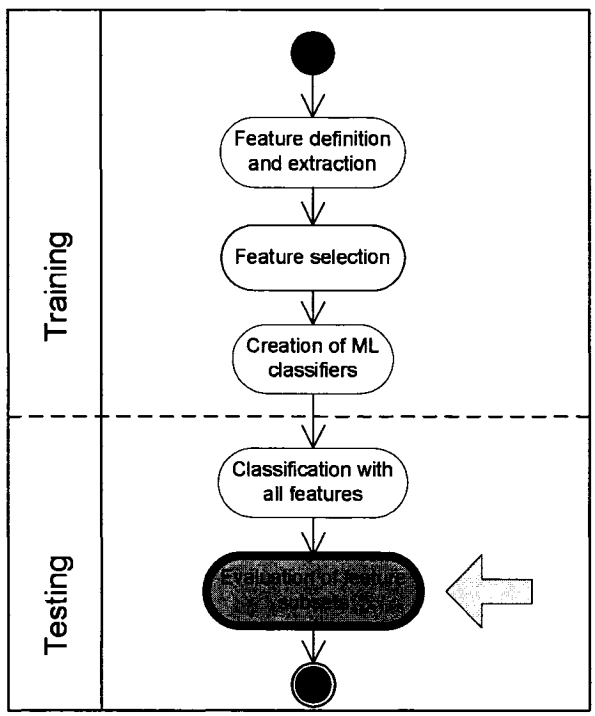


Figure 35: The task of evaluating feature subsets in the set of experiments

The decrease in classification accuracy is considered proportional to the relevancy of the collection of features in the category for genre classification. We begin by removing each feature category from the dataset at a time. We then perform 10 runs of FEATUROMETRE, with a stratified 10-fold cross-validation for each modified version of the feature set.

The gradual removal of all the features of one category with FEATUROMETRE allows the domain expert to evaluate the usefulness of the features within the category. With this approach, a domain expert in future experiments would be able to define better features for a category. Considering that the feature categories in our domain have close relationship to distinct areas of Music Theory, the process can potentially contribute to a better understanding of what characterizes genre and how to best represent musical properties of different areas of Music Theory.

The removal of the instrumentation category affected performance most. The decrease in accuracy is not significant, however. One observation is that the removal of one category of features does not decrease significantly the accuracy.

Applying the findings of filter-based feature selection, we incrementally remove features that scored low. We quantify the effects according to the relative decrease in accuracy and in time for testing taking as reference the accuracy and performance when we tested the complete feature set in the first experiment. Here also 10 runs of FEATUROMETRE with stratified 10-fold cross-validation are performed. Execution time in the experiment is measured immediately before and after each run and then averaged. Two computers with processors dedicated to the experiment were used.

Table 13: Classification accuracy removing features from categories

Feature set	SVM – polynomial kernel		SMV – Gaussian kernel	
	Accuracy	**Usefulness of absent feature category**	**Accuracy**	**Usefulness of absent feature category**
Complete, except distances in the harmonic möbius strip	79.96 %	2.23 %	76.35 %	4.14 %
Complete, except distances in the line of fifths	81.97 %	0.22 %	78.79 %	1.7 %
Complete, except scale	81.87 %	0.32 %	77.62 %	2.87 %
Complete, except rhythmic syncopation and meter	83.13 %	-0.94 %	79.85 %	0.64 %
Complete, except polyphony measurements	82.29 %	-0.1 %	78.47 %	2.02 %
Complete, except duration	82.29 %	-0.1 %	80.06 %	0.43 %
Complete, except instrumentation	77.73 %	4.46 %	74.02 %	6.47 %
Complete	*82.19 %*		*80.49 %*	

Instrumentation is the feature category that is the most useful for both classifiers. The second most useful category is that of distances in the harmonic möbius strip. While instrumentation is directly related to the timbres, the harmonic information arises from the musical content. The fact that these two categories present the most useful features for genre classification by itself is of musicological value. Some categories presented a negative measurement of usefulness with the polynomial kernel. That indicates that the classifier achieved better classification accuracy without the features in each one of these

categories. The category of rhythm syncopation and meter deserves special attention, since rhythm is an important aspect of Music Theory. The result does not necessarily mean that rhythm and meter do not convey useful information for genre determination. On the contrary, two alternatives are possible for higher classification accuracy with the help of rhythmic features. The first alternative is to seek new features based on rhythm. Features to be explored include measurements of auto-similarity. The second alternative considers the fact that the representation of whole rhythmic information of one piece by continuous values is challenging. The alternative would seek classifiers that allow a more direct comparison of the rhythmic contents of two given instances.

Table 14: Removal of low-ranked features

Feature set	Reduction in the number of features	Usefulness of absent features		Decrease in execution time	
		SVM - Polynomial kernel	SVM - Gaussian kernel	SVM - Polynomial kernel	SVM - Gaussian kernel
Without lowest 2 in rank	3.13 %	0.13 %	0.00 %	2.04 %	2.95 %
Without lowest 4 in rank	6.25 %	0.13 %	0.00 %	3.80 %	4.28 %
Without lowest 9 in rank	14.06 %	0.39 %	0.00 %	6.06 %	6.85 %

As the usefulness columns in Table 14 show, the removal of features that scored low during feature selection did not improve empirical classification accuracy. The only noticeable benefit from the elimination of these features was the decrease in absolute testing time in the order of seconds. The decrease is proportionally represented so that interpretation and future comparisons do not need to take hardware details into consideration. The decrease in testing time of both classifiers versus their decrease in empirical accuracy is interpreted as a beneficial trade-off. The decrease in accuracy was very small or zero, while the decrease in test execution time for both classifiers was noticeable.

Removing other features individually also alters the empirical accuracy. In the final step of the second experiment, we list the features that changed accuracy the most.

The maximum empirical accuracy achieved with SVMs was with the complete feature set except the syncop-max feature, which was 83.13 %. That is the same accuracy achieved

by removing all the features from the rhythmic syncopation and meter category. The empirical accuracy for the respective periods is 89.5 %.

Table 15: Classification accuracy removing individual features

Feature set	SVM – polynomial kernel		SMV – Gaussian kernel	
	accuracy	Usefulness of absent feature	accuracy	Usefulness of absent feature
Complete, except syncop-max	83.13 %	- 0.94 %	80.70 %	-0.21 %
Complete, except meter-triple	82.71 %	- 0.52 %	80.59 %	-0.1 %
Complete, except meter-duple	82.40 %	- 0.21 %	80.81 %	-0.32 %
Complete, except stddev-polyph	81.76 %	0.43 %	79.00 %	1.49 %

Note that the usefulness of individual features syncope-max, meter-triple and meter-duple are negative with both SVM classifiers. That means that the accuracy is higher when removing these features individually. The stddev-polyph feature was the most useful individual feature with both classifiers. Recall that instrumentation is a category of features, where each instrument is a Boolean feature. That is likely why stddev-polyph is more useful than instrumental features individually.

4.4 Comparisons

A generic and summarized comparison of different aspects of works that investigated symbolic classification into musical genres is included in Table 16. A direct comparison of the accuracies of the different works in genre classification is not possible because of the differences of the chosen taxonomies, datasets, feature sets and algorithms. The following sections, in turn, comment on general differences between the taxonomies and features.

Table 16: Generic and summarized comparison of approaches for genre classification of symbolic music

	Chai & Vercoe	Shan & Kuo	Pérez-Sancho	McKay	Current work
Input data format	Kern and EsAC	MIDI	MIDI	MIDI (transformed to XML)	Humdrum (transformed to XML)
Genre taxonomy	3 folkloric genres	Enya, Beatles, 2 folk songs	Classical, jazz; Gregorian chants, baroque, ragtime	Bebop, jazz soul, swing, rap, punk, country, baroque, modern classical, romantic	14 classical genres
Monophonic/ Polyphonic	Monophonic	Polyphonic	Monophonic	Polyphonic	Polyphonic
Size of dataset	491	≤ 220 [12]	410	950	943
Accuracy	63 %	64% to 84.2 %	94.3 %	86 %	84.10 %
Algorithm	Hidden Markov Model	Frequent itemset and frequent substring	Naïve Bayes classifier	Neural networks and k-nearest neighbour	Various ML algorithms
Number of features	4		2	109	64

4.4.1 Taxonomies

Tzanetakis calls choir, orchestra, piano and string quartet four classical subgenres. Even though it is possible to classify some pieces according to this taxonomy and assuming the

[12] The number is an estimation. Four genres are said to have at most 55 samples each.

common meaning of "classical" outlined in section 2.2.2, each item is not considered a genre in most of Music literature. Moreover, some classical music may be difficult to classify into the four genres, such as most piano concerti by Mozart, Beethoven, Chopin, Liszt, Rachmaninoff and others which may fall into both piano and orchestra genres, or preludes and fugues by J.S. Bach in the Well-Tempered Clavier executed by a harpsichord, which should not fall into any of the four genres as the taxonomy is defined.

Chai and Vercoe restrict the dataset in two important ways: first, only three folkloric genres were considered. While the specificity may have increased the difficulty in discriminating between the genres, it does not represent a wide choice. The second restriction is that only monophonic pieces were considered.

In the work of Shan and Kuo, while melody is not used in isolation, its use in genre classification needs to be used carefully. Melody is one of the characteristics that vary the most within a given genre. Seldom do two melodies coincide or closely look alike in different pieces. The second remark is that *styles* are loosely defined and the criteria for the particular choices are not clear. The approach to chord extraction is a powerful feature of the work. In our case, this has been captured in the distances in the harmonic möbius strip. We also did assign chord labels, deriving the information from notes. In our case, we extended Humdrum's chord analysis tool to fit the chords into the seven degrees, and calculated the distances in the harmonic möbius strip.

Pérez-Sancho does achieve a very high accuracy by using a text categorization approach. It includes few classical genres in its implementation. It is, however, restricted to monophonic pieces in the dataset. For instance, many if not most of J.S. Bach's (Baroque) music is polyphonic. Also, Pérez-Sancho performs tests on a taxonomy of two genres (called *styles* in his terminology) followed by tests on three genres, which form a rather small taxonomy. McKay works describe a hierarchical taxonomy that achieves an accuracy of 86 % in a flat taxonomy with nine varied genres. This work, in turn, achieved 83.13 % accuracy in a flat taxonomy with 14 genres. A direct comparison has to take into consideration the fact that the genres considered are not all classical, however. Moreover, these are more heterogeneous. In McKay's flat nine genre taxonomy, three were Romantic, Baroque and Modern, while the other six were non-classical. McKay's

experiments with a larger taxonomy of 38 genres leads to accuracies ranging from 52 % to 57 %.

Heterogeneity of the taxonomy is an important aspect of previous works to be considered when attempting to compare the results of the current work with others. Confirming intuition, McKay's experiments have demonstrated that the more specific the taxonomy, the more difficult it is to perform classification. In a hierarchical taxonomy, accuracy is poorer when classifying among leaf genres. In fact, our taxonomy is finer-grained than taxonomies of other works.

4.4.2 Features

The definitions of the features also varied greatly among different works. Table 17 summaries the categories of features used. The summary must be viewed with caution. Even though works may share the same categories, their implementations more often than not differ. For instance, McKay's instrumentation makes a distinction between pitched and unpitched instruments and measures the distribution of notes executed by different kinds of instruments, whereas our work only accounts for the presence of the instruments in a piece. Shan & Kuo derive harmonic information from individual notes, whereas McKay provides mostly statistical measurements of vertical intervals in its chordal category. The actual chords and their inversions are not identified. Arpeggiations are not captured in this category either. While chord spellings are derived essentially from intervals, they do not follow automatically; harmonic analysis is necessary for disambiguation. We actually perform full harmonic analysis and consider chord transitions while computing distances in the harmonic möbius strip during the extraction of features of the first category. We also support chord inversions and arpeggiations when identifying chords.

When comparing features, one may take into account the number of features used. Not only is a small set of features desireable for good performance, but for simplicity and greater domain knowledge. McKay's work also considers 109 features available for classification, some of which are 'compound' features (structured features containing other features), whereas our system considers 64 primitive features. McKay's success

rate drops to near 77 % when 70 features are available for classification of the nine genres. The number of remaining features after feature selection is not reported.

Table 17: Summary of elementary feature categories used in previous works of genre classification of symbolic music

	Chai & Vercoe	Shan & Kuo	Pérez-Sancho	McKay	Current work
Instrumentation				Instrumentation	Instrumentation
Polyphony and other note density statistics				Texture	Polyphony measurements
Rhythm and metre				Rhythm	Rhythmic syncopation and metre
Dynamics				Dynamics (loudness)	
Pitche statistics			Pitch intervals	Pitch statistics	Scale
Melody	Melody	Melody		Melody	
Chords		Chords		Chords	
Duration	Duration	Duration	Duration	Duration	Duration
Harmony					Distances in the harmonic möbius strip
Pitch Spaces					Distances in the line of fifths
Contour	Contour				

More quantitative comparisons between feature sets of previous works and current are not possible for two main reasons. Most importantly, no previous work has measured the relevancy of the features in genre classification. For instance, McKay's work states that different feature selection methods do change accuracy. Even though feature selection based on a Genetic Algorithm was employed for the reduction and weighting of the feature set, no measurement was reported. The other works define a small number of features when compared to McKay's and this work, and do not quantify usefulness either. Secondly, different algorithms were used for classification, as mentioned in section 1.2.2.

Chapter 5 Conclusion

1. Through this work, we have addressed the question of how can computers find useful features for the classification of symbolically encoded classical music into genres with the help of Machine Learning. Specifically, we defined a generic procedure – FEATUROMETRE - inspired in wrapper methods that measures the usefulness of feature subsets. The procedure allows the comparison of feature subsets and helps to define features, being particularly relevant to Constructive Induction and a useful tool to domain experts. The application of the procedure is not restricted to the problem of classical genre classification, but can be applied to other problem domains.

2. This is the first documented work that successfully classifies genres of symbolically encoded classical music considering a representative and comprehensive taxonomy. Previous works have only considered more heterogeneous genre taxonomies. More heterogeneous taxonomies are generally easier to classify than less heterogeneous.

3. Seven categories of features were defined: distances in the harmonic möbius strip, distances in the line of fifths, scale, rhythmic syncopation and meter, polyphony measurements, duration and instrumentation. These categories cover important areas of Music Theory. The 64 features were extracted from the KernScores virtual library. When grouped by categories, all of them showed to be useful for classical genre classification with the SVM with Gaussian kernel, while distances in the harmonic möbius strip, distances in the line of fifths, scale and instrumentation showed to be useful with the binary SVM with polynomial kernel.

4. The comparison between features defined in the proposed work and those proposed in previous work cannot be quantitative, since no previous work has documented measurements on feature subsets or feature ranking. General comparisons are made in section 4.4.

5. Two experiments with different machine learning algorithms compared classifiers that learned how to assign genre labels to unseen music pieces. The 10-fold cross-validation methodology separated the data in such a way that the classifiers only tested on unseen music pieces. The best empirical accuracy achieved was 84.10 % with the Baysian network classifier for multiclass classification into genres, while the maximum accuracy when considering a categorization of the periods was 89.81 % with the random forest classifier.

6. The removal of instrumentation from the feature set affected accuracy the most. Instrumentation showed 4.46 % usefulness. Therefore, features in instrumentation may be considered the most influential features for classical genre determination in this work. The conclusion is consistent with other reports such as McKay's, who highlights the highest internal weighting of 47% to instrumentation features in his classification problem. Unfortunately, this is the only comment in McKay's works to date with a quantitative measurement of a feature category. With the aid of the FEATUROMETRE procedure, we are able to relate the two outcomes.

7. The feature selection step ranked features by weighting five filter-based methods. Removing the lowest ranking features resulted in minor a change in accuracy and in greater gain in execution time performance in classification. The outcome is consistent with what is expected from a feature evaluation procedure.

8. With the proposed procedure we could verify that feature of rhythmic syncopation and meter information unfortunately are not capturing relevant information for better genre determination. The variety of time signatures and rhythmic patterns may be diverse to the point that the kind of syncopation measurements included is not sensitive to different genres. More likely, representing the rhythmic characteristics of one piece with one real value and comparing with the real value of a second piece is not an expressive way to measure the possible rhythmic co-similarities. A more direct comparison of the whole rhythmic information between the pieces is suggested. The result does not necessarily mean that metric alignment is not necessary; on the contrary, better features that maintain metric alignment can be defined in the future.

5.1 Future Research

Future research can conduct a thorough musicological interpretation of the features and their correlation to genres. One suggestion for the assessment of features and datasets would be to define and fix benchmark learning machines. Another research endeavour may try to minimize the number of features, while analyzing the effects on accuracy.

While genre has been choosen as the target category feature, future research may classify music pieces by authorship for instance. The FEATUROMETRE procedure could be used to compare the usefulness of feature categories for the problems of genre and author classification. Classifiers could be combined hierarchically, exploring the taxonomy including periodic information. In this arrangement, a set of classifiers could be trained to classify according to the period of the music, while another set of classifiers would make a more specific genre classification within the different periods. The hybrid taxonomy would still be different than other published hierarchical taxonomies that contain only genres (Pachet & Cazaly, 2000) (C. McKay, 2004).

Better rhythmic measurements may be able to improve the accuracy performance of the classifiers. One approach would be to perform vertical partitions in quantization units of the pieces and compare directly the existence of onsets in the time slots. Variations that can be further explored are the concepts of rhythmic self-similarity and architectonic levels. The idea is to explore the fact that certain rhythmic patterns may be found in different parts of the music possibly modified, being proportionally amplified or reduced, usually by a factor of 2. The two approaches would allow a direct rhythmic comparison between two pieces, instead of trying to express in one feature value the rhythmic richness of one piece.

Intervallic information is not currently captured in any of the seven proposed feature categories. How much redundancy intervallic features would introduce with chordal features already present can be investigated.

In future work, it is desirable that the parametric search in the inner cross-validation step of the first experiment is automated. One approach would be to seek initial parametric

values based on prior information from the data. Then, the classifier could be wrapped and search heuristics applied for iterating through the parameter space.

The taxonomy could be expanded to include modern and post-modern periods. Impressionism, serialism and aleatoric music would be important genres to be considered. The abandonment of tonality in these genres would probably lessen the relevance of features of the harmonic category. The representation of music and feature definition for other genres in modernism such as experimentalism and microtonal music could also be explored.

One could investigate the combination of classifiers that compute on monophonic partitions of the music using classifier selection, considering there have been good classification results achieved by Pérez-Sancho on monophonic music, albeit the small genre taxonomy of this particular case. The challenge with this approach would be that no data that rely on polyphony such as vertical intervals, harmony, texture and counterpoint would be available to each classifier. Research could investigate if there can be more efficient classifiers computing on monophonic data combined by selection that outperform one classifier that takes polyphonic music as input.

A future experiment might perform encodings of Schenkerian analysis of certain pieces, looking at differences in feature evaluation and classification accuracy before and after the Schenkerian analysis. Schenkerian analysis reduced the music using special notation, producing what Schenker called the *Ursatz*. The Ursatz tries to capture basic harmonic and melodic structure of a piece. The objective of the experiment would be to try to perform computations on the original pieces and on the Schenkerian Ursatzs, analyzing the results.

Appendix A – Ranking of Features

Rank	Gain ratio	Information gain	Chi2	RELIEF	1R	Normalised score
1	lvox	avg-polyph	avg-polyph	lbass	avg-polyph	avg-polyph
2	lcor	max-polyph	max-polyph	lcalto	duration	max-polyph
3	ltimpa	duration	lvox	lsoprn	max-polyph	lvox
4	ltromp	stddev-polyph	stddev-polyph	ltenor	lbass	stddev-polyph
5	lfagot	acc-moeb-dist	lvioln	lorgan	lcalto	lbass
6	lflt	acc-moeb-ton-dist	rlofcog-adeviation	meter-duple	lsoprn	lcalto
7	lcbass	lbass	duration	meter-triple	ltenor	lsoprn
8	loboe	lcalto	lviola	lpiano	acc-moeb-dist	ltenor
9	Shexatonic	lsoprn	loboe	lvioln	rlofcog-adeviation	duration
10	lclars	ltenor	lcor	avg-polyph	stddev-polyph	lvioln
11	syncop-exception	rlofcog-adeviation	ltimpa	stddev-polyph	acc-moeb-ton-dist	acc-moeb-dist
12	Spentatonic	lvioln	ltromp	max-polyph	lvioln	lcor
13	Schromatic	lvox	lfagot	lcello	lorgan	ltimpa
14	lviola	lpiano	lflt	acc-moeb-dist	meter-duple	ltromp

15	meter-irregular	lorgan	acc-moeb-dist	lcor	meter-triple	lfagot
16	lbass	lcello	lcello	ltimpa	rpcf10	lflt
17	lcalto	lviola	lcbass	ltromp	rpcf8	loboe
18	lsoprn	meter-duple	acc-moeb-ton-dist	lfagot	rpcf4	acc-moeb-ton-dist
19	ltenor	meter-triple	syncop-exception	loboe	ton-moeb-dist-me	lorgan
20	lcello	moeb-dist-me	Schromatic	lviola	syncop-max	lpiano
21	lvioln	syncop-mean	lpiano	lflt	lcello	lviola
22	lpiano	ton-moeb-dist-me	Shexatonic	acc-moeb-ton-dist	rlofcog	meter-duple
23	lcemba	loboe	lclars	rpcf10	lpiano	lcbass
24	avg-polyph	rpcf10	meter-irregular	rpcf3	syncop-mean	lcello
25	duration	lcor	lbass	syncop-max	moeb-dist-me	rlofcog-adeviation
26	max-polyph	ltimpa	lcalto	rpcf9	rpcf2	meter-triple
27	stddev-polyph	ltromp	lsoprn	lcbass	rpcf11	syncop-exception
28	lorgan	lfagot	ltenor	rpcf4	lvox	Schromatic
29	acc-moeb-dist	lflt	rpcf4	ton-moeb-dist-me	rpcf6	lclars
30	meter-duple	lcbass	Spentatonic	moeb-dist-me	rpcf3	Shexatonic

31	meter-triple	rpcf4	lcemba	rpcf8	lcemba	meter-irregular
32	acc-moeb-ton-dist	lcemba	syncop-mean	lvox	rpcf7	Spentatonic
33	rlofcog-adeviation	rpcf6	lorgan	duration	ton-moeb-dist-chng	ton-moeb-dist-me
34	Sheptatonic	rpcf1	ton-moeb-dist-me	rlofcog-adeviation	moeb-dist-chng	moeb-dist-me
35	syncop-mean	rpcf3	meter-duple	rpcf5	lcor	lcemba
36	ton-moeb-dist-me	syncop-max	meter-triple	Schromatic	ltimpa	rpcf10
37	syncop-max	lclars	moeb-dist-me	syncop-exception	ltromp	rpcf4
38	rpcf3	rpcf11	rpcf10	meter-irregular	lfagot	syncop-mean
39	rpcf11	moeb-dist-chng	rpcf1	rpcf1	lflt	syncop-max
40	moeb-dist-me	rpcf0	syncop-max	rpcf7	loboe	rpcf3
41	rpcf6	rpcf8	rpcf11	rpcf2	lviola	rpcf8
42	rpcf1	rpcf7	rpcf6	lclars	lcbass	rpcf1
43	moeb-dist-chng	Schromatic	moeb-dist-chng	rpcf6	lclars	rpcf6
44	rpcf0	meter-irregular	rpcf0	moeb-dist-chng	rpcf5	rpcf11
45	rpcf4	syncop-exception	rpcf3	ton-moeb-dist-chng	rpcf1	moeb-dist-chng

46	rpcf10	Shexatonic	rpcf8	rpcf11	rpcf9	rpcf7
47	rpcf8	Spentatonic	rpcf7	lcemba	syncop-exception	rpcf0
48	rpcf7	Sheptatonic	Sheptatonic	Shexatonic	Schromatic	rpcf9
49	rpcf9	rpcf9	rpcf9	rlofcog	meter-irregular	Sheptatonic
50	rpcf5	rpcf5	rpcf5	rpcf0	Shexatonic	rpcf5
51	rpcf2	rpcf2	rpcf2	syncop-mean	Spentatonic	rpcf2
52	ton-moeb-dist-chng	ton-moeb-dist-chng	ton-moeb-dist-chng	Spentatonic	lgcass	ton-moeb-dist-chng
53	rlofcog	rlofcog	rlofcog	lgcass	lpiatt	rlofcog
54	lgcass	lgcass	lgcass	lpiatt	ltrngl	lgcass
55	lpiatt	lpiatt	lpiatt	ltrngl	Sheptatonic	lpiatt
56	ltrngl	ltrngl	ltrngl	Sheptatonic	lfltda	ltrngl
57	lfltda	lfltda	lfltda	lfltda	lfltdb	lfltda
58	lfltdb	lfltdb	lfltdb	lfltdb	lfltds	lfltdb
59	lfltds	lfltds	lfltds	lfltds	lfltdt	lfltds
60	lfltdt	lfltdt	lfltdt	lfltdt	llh	lfltdt
61	llh	llh	llh	llh	lrh	Stoofew
62	lrh	lrh	lrh	lrh	Stoofew	lclar
63	Stoofew	Stoofew	Stoofew	Stoofew	lclar	avg-polyph
64	lclar	lclar	lclar	lclar	rpcf0	max-polyph

Bibliography

Alpaydin, E. (2004). *Introduction to machine learning*. Cambridge, Mass.: MIT Press.

Bent, I. D., & Pople, A. (2006). Analysis. In L. Macy (Ed.), *Grove music online*. Retrieved 6 February 2006, from http://www.biblio.uottawa.ca

Blum, A., & Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence, 97*(1-2), 245-271.

Bouckaert, R. R. (2004). *Bayesian Network Classifiers in Weka*. Retrieved 18 May 2006, from http://weka.sourceforge.net/manuals/weka.bn.pdf

Breiman, L. (1996). Technical note: Some properties of splitting criteria. *Machine Learning, 24*(1), 41-47.

Braiman, L. (2001). Random Forests. *Machine Learning, 45*(1), 5-32.

Brown, C. (2006). Ornaments. In L. Macy (Ed.), *Grove music online*. Retrieved 6 February 2006, from http://www.biblio.uottawa.ca

Chai, W., & Vercoe, B. (2001). Folk music classification using hidden markov models. *International conference on artificial intelligence.*

Constitution on the Sacred Liturgy, (1963). Retrieved March 2, 2006, from http://www.vatican.va/archive/hist_councils/ii_vatican_council/documents/vat-ii_const_19631204_sacrosanctum-concilium_en.html

Crammer, K., & Singer, Y. (2001). On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research 2,* 265-292.

Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machines : And other kernel-based learning methods.* Cambridge ; New York: Cambridge University Press.

Fabbri, F. (1999). *Browsing Music Spaces: Categories and the Musical Mind* (University of Surrey, Guildford ed.)

Good, M. (2002). MusicXML in practice: Issues in translation and analysis. *First international conference MAX 2002: Musical application using XML,* Milan, 47-54.

Gray, R. M. (1990). *Entropy and information theory.* New York: Springer-Verlag.

Grove, G., & Sadie, S. (1980). *The new grove dictionary of music and musicians.* London; Washington, D.C.: Macmillan Publishers; Grove's Dictionaries of Music.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research, 3*, 1157-1182.

Hastie, T., & Tibshirani, R. (1998). Classification by pairwise coupling. *Advances in neural information processing systems.*

Holte, R. C. (1993). Very simple classification rules perform well on most commonly used datasets. *Machine Learning, 11*, 63-91.

Huron, D. (2006). *The humdrum toolkit: Software for music research.* Retrieved March 7, 2006 from http://dactyl.som.ohio-state.edu/Humdrum/

Joachims, T. (2004) *SVMLight.* Retrieved May 22, 2006 from http://svmlight.joachims.org/

Jones, G. T. (1974). *Music theory*. New York: Barnes & Noble Books.

Kira, K., & Rendell, L. A. (1992). The feature selection problem: Traditional methods and a new algorithm. Paper presented at the *10th national conference on artificial intelligence,* 129-134.

Knopoff, L., & Hutchinson, W. (1981). Information theory for musical continua. *Journal of Music Theory, 25*(1), 17-44.

Kuncheva, L. I. (2004). *Combining pattern classifiers : Methods and algorithms.* Hoboken, NJ: J. Wiley.

Larue, J., & Wolf, E. K. (2006). Symphony. In L. Macy (Ed.), *Grove music online.* Retrieved 25 February, 2006 from http://www.biblio.uottawa.ca

Liu, H., & Setiono, R. (1995). Chi2: Feature selection and discretization of numeric attributes. *7th IEEE international conference on tools with artificial intelligence,* 88.

Lo, S., & Famili, A. (1997). Development of a knowledge-driven constructive induction mechanism. *Proceedings of the Second International Symposium on Intelligent Data Analysis,* London, UK.

Longuet-Higgins, H. C., & Lee, C. S. (1982). The perception of musical rhythms. *Perception, 11*, 115-128.

Longuet-Higgins, H. C. (1994). Artificial intelligence and musical cognition. *Philosophical Transactions: Physical Sciences and Engineering, 349*(1689), 103-113.

Mangsen, S., Irving, J., Rink, J., & Griffiths, P. (2006). Sonata. In L. Macy (Ed.), *Grove music online.* Retrieved 6 February 2006, from http://www.biblio.uottawa.ca

Mazzola, G., Göller, S., & Müller, S. (2002). *The topos of music : Geometric logic of concepts, theory, and performance.* Boston, MA: Birkhauser Verlag.

McKay, C., & Fujinaga, I. (2004). Automatic genre classification using large high-level musical feature sets. *International conference on music information retrieval,* 525.

McKay, C. (2004). Automatic genre classification of MIDI recordings. (Masters Thesis, McGill University).

Merryman, M. (1997). *The music theory handbook.* Fort Worth, Texas: Harcourt Brace College Pub.

Meudic, B. (2003). Musical similarity in a polyphonic context : A model outside time. *XIV colloquium on musical informatics,* Firenze, Italy.

Pachet, F., & Cazaly, D. (2000). A taxonomy of musical genres. *Content-based multimedia information access (RIAO),* Paris, France.

Pérez-Sancho et al. (2005). A text categorization approach for music style recognition. *Pattern recognition and image analysis,* Estoril, Portugal, *3523* 649-657. from http://uclibs.org/PID/96891; http://uclibs.org/PID/96892

Platt, J. C. (1999). Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. J. C. Burges & A. J. Smola (Eds.), *Advances in kernel methods : Support vector learning*. Cambridge, Mass.: MIT Press.

Pye, D. (2000). *Content-Based Methods for the Management of Digital Music*. Piscataway, New Jersey: Ieee.

Quinlan, J. R. (1986). Induction of Decision Trees. *Machine Learning* (1), 81-106.

Quinlan, J. R. (1993). *C4.5 : Programs for machine learning*. San Mateo, Calif.: Morgan Kaufmann Publishers.

Randel, D. M. (2003). *The harvard dictionary of music* (4th ed.). Cambridge, Mass.: Belknap Press of Harvard University Press.

Rudziński, W., & Moffa, R. (1993). *Il ritmo musicale : Teoria e storia*. Lucca: Libreria musicale italiana.

Samson, J. (2006). Genre. In L. Macy (Ed.), *Grove music online*. Retrieved February 21, 2006 from http://www.bib.uottawa.ca

Sapp, C. S. (2005). *KernScores*. Retrieved March 7, 2006 from http://kern.humdrum.net/

Schaffrath, H. (1997). The essen associative code: A code for folksong analysis. In E. Selfridge-Field (Ed.), *Beyond MIDI : The handbook of musical codes* (pp. 343-361). Cambridge, Mass.: MIT Press.

Selfridge-Field, E. (1997). *Beyond MIDI : The handbook of musical codes*. Cambridge, Mass.: MIT Press.

Shan, M., Kuo, F., & Chen, M. (2003). Music style mining and classification by melody. *IEICE transactions on information and systems,* 655-659.

Shawe-Taylor, J., & Cristianini, N. (2004). *Kernel methods for pattern analysis.* Cambridge, UK ; New York: Cambridge University Press.

Sleator, D., & Temperley, D. (2006). *The melisma music analyzer.* Retrieved March 07, 2006 from http://www.link.cs.cmu.edu/music-analysis/

Smither, H. E. (1977). *A history of the oratorio.* Chapel Hill: University of North Carolina Press.

Steedman, M. (1984). A generative grammar for jazz chord sequences. *Music Perception, 2*, 52-77.

Straub, H. (2004). *Music Theory/Music and mathematics: Frequently asked questions.* Retrieved March 10, 2006 from http://home.datacomm.ch/straub/mamuth/mamufaq.html

Temperley, D. (2000). The line of fifths. *Music Analysis, 19*(3), 289-319.

Tzanetakis, G., & Cook, P. (2002). Musical genre classification of audio signals. *IEEE transactions on speech and audio processing, 10*(5) 293-302.

Vignal, M. (1987). *Dictionnaire de la musique.* Paris: Larousse.

Walker, P. (2006). Fugue. In L. Macy (Ed.), *Grove music online.* Retrieved February 25, 2006 from http://www.bib.uottawa.ca

WEKA. Retrieved May 22, 2006 from http://www.cs.waikato.ac.nz/ml/weka/

Wikipedia contributors. (2006). *Musical Form*. Retrieved February 23, 2006 from http://en.wikipedia.org/wiki/Main_Page

Xu, C., Maddage, N. C., Shao, X., Cao, F., & Tian, Q. (2003). Musical genre classification using support vector machines. *International conference of acoustics,speech & signal processing,* HongKong,China.

Zamacois, J. (1960). *Curso de formas musicales*. Barcelona: Editorial Labor.